

FACOLTÀ DI INGEGNERIA

LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

“Modellazione dei processi di business relativi alla produzione degli ortaggi, in accordo allo standard ebXML, e validazione di una piattaforma di tracciabilità alimentare

RELATORI

IL CANDIDATO

Prof. Beatrice Lazzerini
Dipartimento di Ingegneria dell'Informazione

Innesti Marco

Prof. Francesco Marcelloni
Dipartimento di Ingegneria dell'Informazione

Prof. Andrea Tomasi
Dipartimento di Ingegneria dell'Informazione

Anno Accademico 2005-2006

INDICE

| | |
|--|-----------|
| INDICE DELLE FIGURE..... | 3 |
| INTRODUZIONE | 5 |
| CAPITOLO 1 – LA TRACCIABILITÀ DI FILIERA | 6 |
| 1.1 La tracciabilità di filiera ed i suoi servizi..... | 6 |
| 1.2 Metodo organizzativo per la tracciabilità..... | 9 |
| 1.3 Le unità fondamentali dei processi di tracciabilità | 12 |
| CAPITOLO 2 – ANALISI DI UNA FILIERA REALE E RELATIVA | |
| MODELLAZIONE..... | 15 |
| 2.1 Introduzione | 15 |
| 2.2 Sintesi dei processi di produzione del Cavolo Nero | 16 |
| 2.3 Introduzione a UML..... | 20 |
| 2.4 Introduzione al modello Cerere..... | 25 |
| 2.5 Modelli dinamici per la filiera del cavolo nero | 28 |
| 2.6 Modelli statici per la filiera del cavolo nero | 34 |
| 2.7 Disciplinare semplificato delle procedure in filiera | 46 |
| 2.7.1 Attore Responsabile - Coordinatore | 46 |
| 2.7.2 Coltivatore – Fornitore Piante | 47 |
| 2.7.3 Coltivatore - Confezionatore | 48 |
| 2.7.4 Confezionatore - Distributore | 49 |
| CAPITOLO 3 – LO STANDARD ebXML | 51 |
| 3.1 L'architettura ebXML | 51 |
| 3.1.1 Elementi principali di ebXML | 52 |
| 3.1.2 Business Process | 53 |
| 3.1.3 Collaboration Protocol Profile | 54 |
| 3.1.4 Collaboration Protocol Agreement | 55 |
| 3.1.5 Message Service | 57 |
| 3.1.6 Registry | 58 |
| 3.1.7 Core Components | 58 |
| 3.1.8 Il funzionamento di ebXML..... | 59 |
| 3.2 Modelli ebXML | 61 |
| 3.2.1 Analisi specifica dei campi dei Business Process | 61 |

| | |
|--|------------|
| 3.2.2 Analisi specifica dei CPA | 64 |
| 3.3 Come utilizzare i documenti prodotti..... | 70 |
| 3.3.1 ebMail | 71 |
| 3.3.2 Plug-In per ebMail | 74 |
| CAPITOLO 4 – PRESENTAZIONE DEL PROTOTIPO DI SUPPORTO ALLA | |
| TRACCIABILITA’ | 76 |
| 4.1 Il prototipo Cerere | 76 |
| 4.2 Utente Azienda..... | 76 |
| 4.3 Utente Coordinatore | 80 |
| 4.4 Altri utenti | 83 |
| CAPITOLO 5 – VALIDAZIONE DEL SISTEMA INFORMATICO | 87 |
| 5.1 Finalità della validazione | 87 |
| 5.2 I Pacchetti Informativi..... | 89 |
| 5.3 Ricezione dei messaggi | 92 |
| 5.4 Verifica dell’avvenuta tracciatura dei lotti..... | 94 |
| CONCLUSIONI | 100 |
| APPENDICE A | 102 |
| APPENDICE B | 105 |
| APPENDICE C | 108 |
| APPENDICE D | 110 |
| APPENDICE E | 117 |
| APPENDICE F | 118 |
| APPENDICE G | 120 |
| APPENDICE H..... | 122 |
| APPENDICE I | 141 |
| APPENDICE L | 159 |
| BIBLIOGRAFIA | 166 |

INDICE DELLE FIGURE

| | |
|--|----|
| Figura 1 Rintracciabilità di lotti potenzialmente pericolosi | 8 |
| Figura 2 Flow sheet di una filiera..... | 10 |
| Figura 3 Interazione tra i principali attori di una filiera | 11 |
| Figura 4 I flussi della filiera | 14 |
| Figura 5 Esempio di diagramma delle classi..... | 21 |
| Figura 6 Esempio di diagramma di sequenza..... | 23 |
| Figura 7 Esempio di diagramma di Attività | 24 |
| Figura 8 Diagramma delle classi del modello per la tracciabilità | 25 |
| Figura 9 Diagramma di stato per l'entità lotto | 26 |
| Figura 10 Diagramma di attività per un lotto | 26 |
| Figura 11 Diagrammi di sequenza per il prototipo cerere | 27 |
| Figura 12 Diagramma di attività Fornitore Piante..... | 29 |
| Figura 13 Diagramma di attività Coltivatore..... | 30 |
| Figura 14 Diagramma di attività Confezionatore | 31 |
| Figura 15 Diagramma di attività Distributore | 33 |
| Figura 16 Struttura statica per Sementi | 34 |
| Figura 17 Struttura statica Balla di sementi..... | 35 |
| Figura 18 Struttura statica per cassa di piante | 36 |
| Figura 19 Struttura statica per cassa di piante ceduta..... | 36 |
| Figura 20 Struttura statica per cassa di piante acquisita..... | 37 |
| Figura 21 Struttura statica per casse di cavoli | 38 |
| Figura 22 Casse di cavoli – caratteristiche lotto..... | 39 |
| Figura 23 Struttura statica per casse di cavoli immagazzinate..... | 40 |
| Figura 24 struttura statica per casse di cavoli cedute | 40 |
| Figura 25 Struttura statica per casse di cavoli acquisite..... | 41 |
| Figura 26 Struttura statica per cavoli condizionati..... | 41 |
| Figura 27 Struttura statica per cavoli calibrati..... | 42 |
| Figura 28 Struttura statica per confezioni di cavoli..... | 42 |
| Figura 29 Struttura statica per confezioni di cavoli conservate..... | 43 |
| Figura 30 Struttura statica per confezioni di cavolo cedute | 43 |
| Figura 31 Struttura statica per confezioni di cavolo acquisite..... | 44 |

| | |
|---|----|
| Figura 32 Struttura statica per confezioni di cavoli conservate..... | 44 |
| Figura 33 Struttura statica per confezioni di cavoli vendute | 45 |
| Figura 34 Diagramma di sequenza ingresso in filiera | 47 |
| Figura 35 Diagramma di sequenza acquisto piante | 48 |
| Figura 36 Diagramma di sequenza vendita cavoli | 49 |
| Figura 37 Diagramma di sequenza vendita confezioni di cavoli | 50 |
| Figura 38 Business collaboration | 54 |
| Figura 39 Tipico scenario di utilizzo di ebXML | 60 |
| Figura 40 Esempio di composizione di delivery channel..... | 67 |
| Figura 41 Interfaccia principale di ebMail | 71 |
| Figura 42 Caricamento di un Plug-in | 72 |
| Figura 43 Configurazione del Software | 73 |
| Figura 44 Invio di un messaggio | 74 |
| Figura 45 Interfaccia per l'invio di un messaggio di Acquisizione..... | 77 |
| Figura 46 Interfaccia per l'invio di un messaggio di Trasformazione | 78 |
| Figura 47 interfaccia per l'invio di un messaggio d Movimentazione | 79 |
| Figura 48 Interfaccia per l'invio di un messaggio di Cessione | 80 |
| Figura 49 Interfaccia per nodo di coordinamento | 81 |
| Figura 50 Registrazione dei cambi di responsabilità..... | 82 |
| Figura 51 Home page del portale WEB | 83 |
| Figura 52 Pagina per la ricerca di un lotto..... | 84 |
| Figura 53 Risultato di una ricerca | 85 |
| Figura 54 Codifica ebXML di un messaggio di acquisizione. | 89 |
| Figura 55 Codifica ebXML di un messaggio di trasformazione. | 90 |
| Figura 56 Codifica ebXML di un messaggio di Movimentazione | 91 |
| Figura 57 Codifica ebXML di un messaggio di Cessione..... | 92 |
| Figura 58 Finestra di ricezione dei messaggi. | 93 |
| Figura 59 Finestra di visualizzazione dei cambi di responsabilità. | 94 |
| Figura 60 Ricerca di un codice. | 95 |
| Figura 61 Schermata 1 dei risultati..... | 96 |
| Figura 62 Schermata 2 dei risultati..... | 97 |
| Figura 63 Schermata del primo cambio di responsabilità. | 98 |
| Figura 64 Cessione al consumatore | 99 |

INTRODUZIONE

Obiettivo della presente tesi è la modellazione di processi produttivi del settore degli ortaggi, in modo più specifico del cavolo nero, e la validazione di un prototipo per la tracciabilità dei prodotti alimentari.

Per prima cosa vengono evidenziati gli aspetti principali e le problematiche relative alla tracciabilità, allo scopo di avere una prima visione di quali siano gli aspetti fondamentali da modellare. Si analizza quindi la filiera da tracciare, introducendo lo Unified Modeling Language (UML), utilizzato per la modellazione, passando infine agli aspetti principali del sistema informativo di tracciabilità *Cerere*, piattaforma sulla quale è stata effettuata la validazione.

La modellazione si sviluppa a partire da una descrizione in linguaggio naturale dei processi produttivi del cavolo nero, ed una successiva identificazione delle trasformazioni rilevanti per la tracciabilità, cioè tali da consentire l'individuazione degli attori responsabili ad ogni stadio della catena produttiva, delle responsabilità, dei lotti generati, dei siti di produzione e delle attività.

Dagli elementi individuati segue la modellazione in notazione UML, con l'obiettivo di ottenere una versione formale delle entità coinvolte e dei processi nei quali si articola la filiera.

Vengono inoltre identificati i processi organizzativi di filiera, quali l'ingresso di una azienda o di un produttore all'interno della filiera stessa.

Alcune di tali procedure vengono poi tradotte in schemi e documenti relativi allo standard ebXML analizzando nel dettaglio le sezioni riservate al corretto svolgimento delle transazioni e le potenzialità offerte dallo standard medesimo.

Viene presentato inoltre un possibile utilizzo applicativo di tali documenti attraverso un opportuno pacchetto software (ebMail) in grado di consentire l'esecuzione di transazioni di business tramite un'interfaccia utente simile a quella di un client di posta elettronica.

La tesi si conclude attraverso una simulazione, sulla piattaforma *Cerere*, del modello di filiera del cavolo nero precedentemente ottenuto, al fine di validare la piattaforma ed il modello medesimo.

Capitolo 1

La tracciabilità di filiera

Questo primo capitolo intende spiegare il concetto di “tracciabilità di filiera”, evidenziandone elementi e strumenti per realizzare un sistema di tracciabilità. E’ l’approccio per conseguire le conoscenze fondamentali utili a comprendere e delineare l’ambito nel quale si svilupperà il lavoro oggetto della presente tesi.

1.1 La tracciabilità di filiera e i suoi servizi

La tracciabilità di filiera è intesa come costante monitoraggio di un dato prodotto all’interno del processo di produzione, dalla materia prima al consumatore.

Da essa si evince quindi la storia di un prodotto, identificando sia le aziende contribuenti alla produzione sia le componenti in ogni singola fase della catena produttiva.

Più precisamente, con il termine di tracciabilità, si intende la possibilità di seguire la storia di un prodotto partendo dalla materia prima fino ad arrivare al prodotto finito.

La rintracciabilità è invece il processo opposto, grazie al quale è possibile risalire dal prodotto finito alle materie prime utilizzate per comporlo.

Tracciabilità e rintracciabilità di filiera rappresentano, per le aziende, un ulteriore impegno in termini di onestà, professionalità e garanzia in rispetto della sempre maggiore richiesta di qualità e trasparenza da parte del consumatore.

La fiducia tra i due estremi del mercato è il fine ultimo della tracciabilità. Quest’ultima, se ben evidenziata, suscita “appeal” e rappresenta un importante strumento di promozione e valorizzazione del prodotto stesso. Anche se risulta complessa (il prodotto passa attraverso aziende di produzione, trasformazione e commercializzazione) la filiera e’ quanto di meglio le aziende possano offrire in termini di garanzia sulla qualità del prodotto, cercando di

ristabilire quella spontaneità del rapporto produttore-consumatore che temevamo essersi dissolto con l'affermarsi dell'era delle macchine.

La tracciabilità di filiera propone un modello di integrazione verticale tra le aziende.

È essenziale, quindi, per tracciare la storia di un prodotto, l'identificazione dei nomi delle aziende partecipanti al sistema produttivo ed il loro grado di responsabilità all'interno del processo produttivo. Sul piano giuridico e gestionale tutto ciò appare importantissimo, anche per le recenti disposizioni di legge della Comunità europea (regolamento CE 178-2002) in vigore dal gennaio 2005, poiché consente la tempestiva individuazione di casi di non conformità oppure l'isolamento di situazioni di rischio per il consumatore, permettendo di individuarne le cause.

Nella figura 1 è esemplificata l'utilità della tracciabilità nel caso dell'identificazione di un lotto contaminato.

È possibile ripercorrere facilmente la filiera a ritroso per identificare cosa ha causato la non conformità e rintracciare tutti e soli gli altri lotti che hanno subito le stesse trasformazioni e risultano quindi a rischio.

I cerchi rappresentano nuovi lotti, mentre i rettangoli, le attività che vanno a comporre tali lotti. Le linee rappresentano il legame tra i lotti di origine ed i lotti generati, quindi le attività con N linee di ingresso stanno a significare una composizione di più ingredienti per formare un nuovo prodotto, mentre le attività con N linee di uscita simboleggiano una suddivisione del lotto originale in più lotti figli.

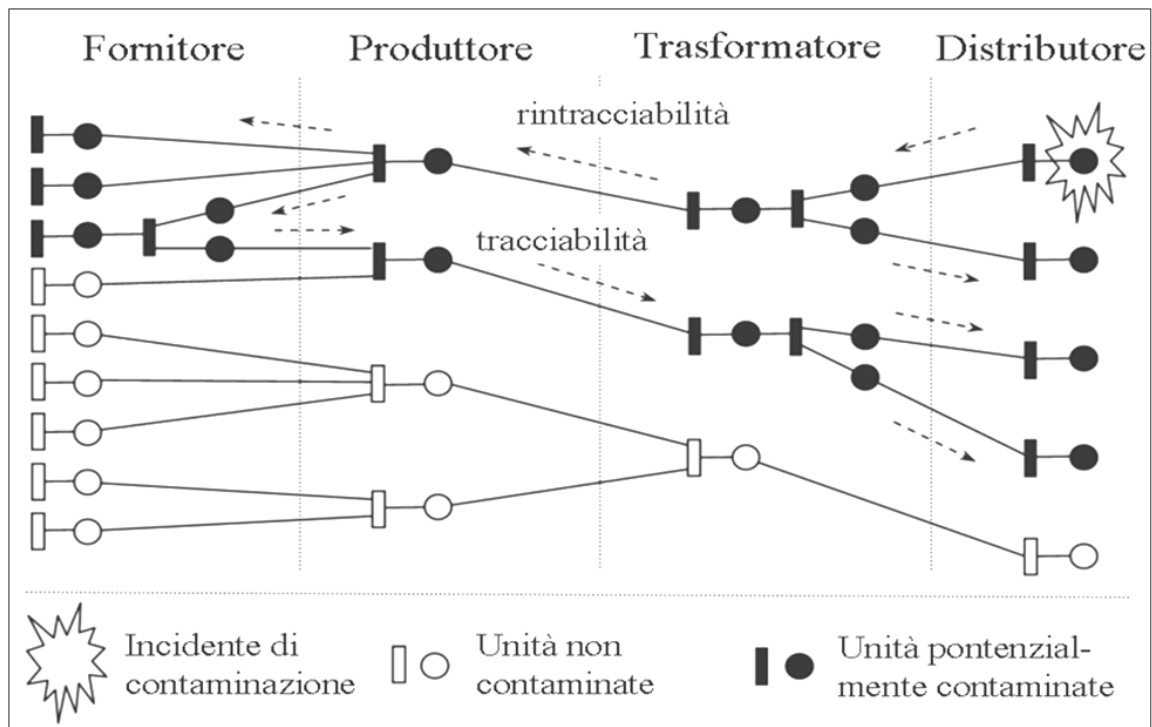


Figura 1 Rintracciabilità di lotti potenzialmente pericolosi

Risulta però presto chiara l'impossibilità di tracciare nelle rispettive filiere, elementi secondari come additivi, coadiuvanti, imballaggi... Il perchè risulta semplice: il sistema diventerebbe troppo complesso, ingestibile.

Tale sistema di controllo del processo produttivo è regolato da un disciplinare, reso maggiormente visibile al consumatore apponendo un logo in etichetta o sulla confezione del prodotto.

All'interno della tracciabilità di filiera possiamo notare che il monitoraggio flussi materiali è uno strumento di controllo dei processi che permette di razionalizzare e isolare la cause di errore.

In questo senso è evidente l'importanza di eseguire sin dall'inizio un'identificazione dei processi critici e una modellazione delle procedure per evitare costi elevati di gestione informativa sulla tracciabilità.

Un sistema di tracciabilità offre molteplici servizi che riguardano sia i consumatori sia le imprese.

- *Al consumatore:* sicurezza alimentare, intesa come garanzia degli ingredienti e dei processi che essi hanno subito, con il ritiro dei prodotti dal mercato in caso di emergenza e dunque, anche conseguente protezione della salute pubblica; prevenzione delle frodi; controllo delle malattie che gli animali possono trasmettere all'uomo; controllo della salute degli animali; aumento delle garanzie sulla conformità del prodotto.

- *Alle imprese:* adeguamento alla legislazione; capacità di adottare azioni rapide per ritirare dal commercio prodotti pericolosi e salvaguardare l'immagine aziendale; riduzione dei costi per un eventuale ritiro di prodotti dal commercio; possibilità di differenziarsi sul mercato; possibilità di garantire la veridicità delle informazioni; diminuzione dei costi dell'organizzazione interna della filiera; agevolazione della scelta dei fornitori e il monitoraggio dei clienti.

1.2 Metodo organizzativo per la tracciabilità

La realizzazione di un sistema di tracciabilità della filiera deve definire il proprio flow-sheet, inteso come identificazione delle aziende, del percorso della materia prima principale e delle aziende fornitrici degli elementi secondari maggiormente rappresentativi.

Nella figura 2 è visibile un esempio di un flow-sheet di filiera in cui vengono rappresentate tutte le aziende che vi partecipano ed i ruoli che rivestono. In tonalità più scura sono rappresentate le aziende che partecipano alla formazione di un prodotto specifico (come dovrà risultare poi dal processo di tracciabilità)

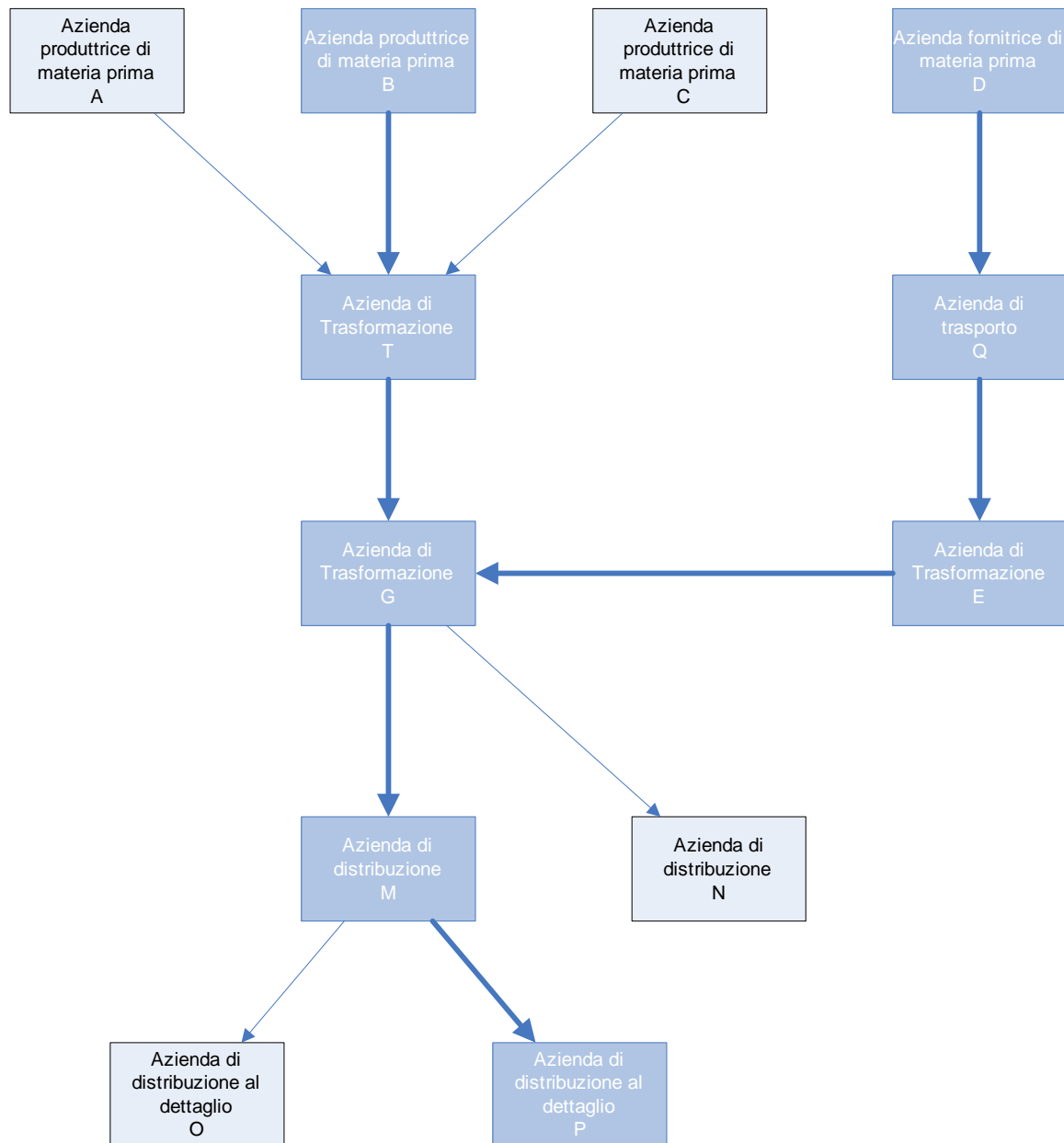


Figura 2 Flow sheet di una filiera

Infatti, risulta logico che non tutte le aziende devono partecipare alla formazione di ogni singolo prodotto, tuttavia è necessario definire un modello organizzativo della filiera e ottenere l'assenso di tutte le aziende che vi partecipano.

Occorre inoltre individuare una figura che organizzi e raccolga le informazioni determinanti ai fini della tracciabilità : questa figura e' il nodo di coordinamento.

Quest'ultimo è autorizzato dalle aziende alla raccolta dei dati ed al controllo del rispetto delle regole della filiera.

Questo è anche l'organo preposto per rispondere alle richieste dei consumatori e dell'autorità sanitaria.

Il nodo di coordinamento è il supervisore delle attività svolte dalle aziende all'interno della Filiera e gestisce il sistema di rintracciabilità e delle modalità e responsabilità per la gestione dei dati e della documentazione di processo.

Alla base di un sistema di rintracciabilità deve esserci un accordo tra le varie aziende e il nodo di coordinamento detto “patto di filiera”

La struttura del patto di filiera è costituita da alcuni documenti fondamentali:

Disciplinare tecnico: riporta le regole delle procedure della filiera.

Sistema documentale: supporta il disciplinare tecnico tramite manualistica e modulistica

Diagramma di flusso: identifica i flussi materiali attraverso le aziende

Piano dei controlli: contiene tipi e modalità dei controlli.

Il sistema di tracciabilità dovrebbe essere sottoposto ad una disciplina di autocontrollo e di certificazione volontaria e può ben operare solamente grazie alla totale interazione tra i tre principali fattori: le aziende della filiera, il nodo di coordinamento, l'organismo di certificazione



Figura 3 Interazione tra i principali attori di una filiera

Naturalmente l'adesione di una azienda ad una filiera è formalizzata da una dichiarazione d'intenti sottoscritta dalla stessa azienda, ed autorizza il nodo di coordinamento ad esporre in chiaro il nome ogni volta che un utente, pubblico o privato, richieda qualsiasi tipo di informazione sulla tracciabilità

1.3 Le unità fondamentali dei processi di tracciabilità

Le masse di prodotto, all' interno della filiera, sono gestite in parti rintracciabili in senso strettamente fisico in punti determinati della filiera stessa. Queste parti sono dette lotti di tracciabilità o di identificazione (da non confondere con lotti logistici).

È importante ricordare che è necessaria solo l'identificazione dell'azienda fornitrice; ne consegue che possono essere considerate lo stesso lotto, le masse contenute in contenitori diversi, a condizione che risultino con caratteristiche omogenee, quindi provenienti dal medesimo processo produttivo.

I punti critici di gestione dei lotti di identificazione in ingresso e uscita devono essere controllati mediante la registrazione di: data, identificazione del prodotto, nome di azienda e siti, massa totale del lotto considerato.

Alla fine, dunque, all' interno di un sistema di tracciabilità devono essere necessariamente identificati i principali elementi costitutivi, sintetizzabili nelle operazioni di identificazione e registrazione dei lotti, nella consapevolezza dei legami tra gli stessi e nella comunicazione tra gli operatori della filiera.

IDENTIFICAZIONE

Consiste nell' identificare lotti che hanno subito precedentemente il medesimo processo di trasformazione. La gestione di tali processi produttivi deve avvenire per lotti in quanto la tracciabilità si riferisce ad un prodotto fisicamente identificabile; questo per rendere possibile il tempestivo riconoscimento delle aziende che hanno contribuito al processo di produzione. Gestire la tracciabilità significa, oltre ad attribuire un'unica identificazione a ciascun raggruppamento, anche seguirne il percorso sino alla vendita al dettaglio, sino al consumatore. Risulta quindi fondamentale, per ogni singola azienda, il contributo delle tecnologie di gestione, intese come capacità di leggere e produrre informazioni in formato standard, ed il metodo di trasferimento delle informazioni, inteso come garanzia di compatibilità e di coerenza delle informazioni gestite.

REGISTRAZIONE

Gestire la tracciabilità significa scegliere quali informazioni registrare all'interno della totalità del percorso compiuto della filiera. È necessario registrare le sole informazioni che consentono di ricostruire:

- I flussi in entrata
- Il processo di trasformazione
- i flussi in uscita

Le normative vigenti obbligano le aziende ad indicare sull'etichetta al consumo alcune informazioni e dunque di registrare e gestire tali informazioni all'interno dei sistemi aziendali.

Ulteriori informazioni, relative all'origine oppure alla composizione sono strettamente arbitrarie e decise dall'azienda stessa al fine di qualificare commercialmente il prodotto.

Stabilire quali e quante debbano essere le informazioni da tracciare è uno degli aspetti più delicati dell'intero processo. La fonte principale per determinare questa scelta è il consumatore.

Va anche considerato che il numero di informazioni da tracciare influenza notevolmente la dimensione di un lotto: tutti i materiali che lo compongono dovranno presentare caratteristiche omogenee rispetto a tali informazioni.

La crescita del numero delle informazioni aumenta la complessità del sistema e il maggior numero di informazioni può condizionare la composizione dei lotti, l'aumento dei processi di revisione di procedure aziendali e, naturalmente, i costi di produzione.

È chiaro, infine, che il sistema di tracciabilità di filiera presuppone la registrazione aggiornata, archiviata e facilmente disponibile delle informazioni sulle attività e sui flussi del processo produttivo.

LEGAMI

La registrazione dei legami tra i lotti e le unità logistiche successive nel corso della trasformazione è determinante.

All'interno di una azienda solo una capillare gestione dei legami permette di risalire alla perfetta connessione tra merce entrata e in uscita. L'impresa è responsabile dei legami tra propri fornitori e propri clienti. Se tali passaggi non sono gestiti in maniera adeguata da un qualsiasi degli attori della filiera, si parla allora di perdita della tracciabilità.

Ci sono tre tipi di legami per monitorare i prodotti nell'interezza dei processi di trasformazione

- 1) **Legami tra lotti di produzione:** riguardano le materie prime, imballi e prodotti finiti.

2) **Legami tra lotti di produzione e unità logistiche:** può accadere che le unità logistiche non vengano assemblate a conclusione della catena produttiva, e dunque si renda necessario registrare i legami di stoccaggio e di logistica

3) **Legami tra unità logistiche:** è il quadro, molto complesso, delle unità logistiche ricevute e spedite

COME INTERAGISCONO GLI OPERATORI DELLA FILIERA

Il fulcro della rintracciabilità di filiera è costituito dal processo di comunicazione, garante dell'intero sistema.

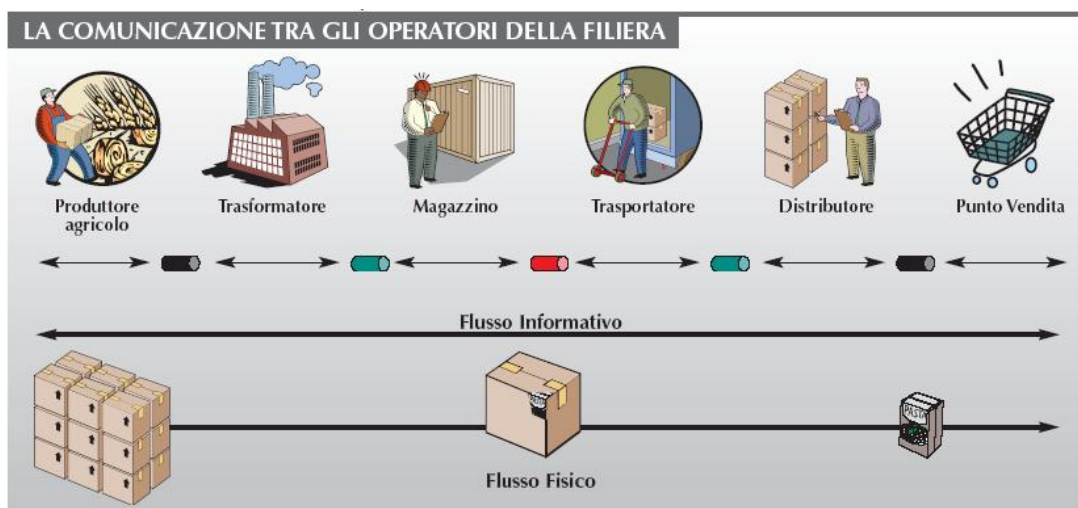


Figura 4 I flussi della filiera

La continuità del flusso di informazioni, che può avvenire in modo cartaceo, elettronico o mediante etichettatura, è garantito dalla specificità dei compiti dei singoli operatori: ognuno di essi deve comunicare le caratteristiche (identificativi) dei lotti tracciati all'operatore successivo, per permettergli di applicare a sua volta i principi alla base della rintracciabilità. Il processo è dunque inter-aziendale: per garantire il successo del sistema di tracciabilità è necessario adottare uno standard condiviso da tutti. Tale scelta presenta numerosi vantaggi tra i quali il rafforzamento del sistema di tracciabilità, ovviamente, ma anche l'interoperabilità e l'indipendenza tecnologica rispetto ad un dato service provider; maggiore efficienza del sistema logistico e sensibile diminuzione dei costi di implementazione.

Capitolo 2

Analisi di una filiera reale e relativa modellazione

Questo capitolo chiarisce l'utilizzo degli strumenti e i modelli scelti per sviluppare la modellazione oggetto della tesi.

Contiene anche una descrizione del linguaggio UML e la spiegazione particolareggiata del modello di riferimento.

2.1 Introduzione

Ora, tenendo presente le problematiche emerse nel capitolo precedente, cercheremo di analizzare un caso reale di filiera agroalimentare, per cercare di applicarvi un sistema di tracciabilità.

Per analizzare la filiera ed estrarne gli attori ed i processi fondamentali, ci siamo basati su alcuni documenti quali il “Manuale di corretta prassi per la produzione del cavolo broccolo e verza” del parco tecnologico agroalimentare dell’Umbria [7] e sulle “schede di produzione e post-raccolta” della Regione Toscana (L.R. 25/99)[8].

Partiremo fornendo una descrizione in linguaggio naturale, cercando di mettere in luce tutti gli aspetti essenziali ai fini della tracciabilità, riguardanti la realtà degli ortaggi, a partire dal campo fino alla distribuzione al consumatore finale.

Questi aspetti ci porteranno poi alla formulazione di un modello formale, basato su UML.

Vengono inoltre presentati i modelli su cui è basato il prototipo Cerere ed al quale noi faremo riferimento per la nostra progettazione e validazione.

2.2 Sintesi dei processi di produzione del cavolo nero

Il cavolo nero è una pianta a ciclo biologico biennale ma normalmente annuo in agricoltura. Ha un apparato radicante che raggiunge i 60 centimetri, uno stelo eretto e legnoso alla base. Le foglie sono semplici, più o meno bollose ed increspate, di colore verde scuro o violacee, con coste bianche più o meno sottili; le foglie larghe involucranti vanno a formare una testa compatta, di forma sferica più o meno schiacciata, con peso da 0,5 a 3 Chilogrammi che rappresenta la parte commestibile.

La temperatura ottimale di germinazione è di 20° C mentre la temperatura base per la crescita vegetativa è pari a 5°C con valori ottimali compresi tra 18° e 22° C.

Si adatta a diversi tipi di terreno; tuttavia predilige quelli a medio impasto con PH ottimale tra 6.5 e 7.5.

Per esigenze dovute a rischi di erosione è vietato coltivare queste piante i terreni con pendenza media superiore al 15%

È inoltre vietata la monosuccessione e quindi il ritorno della coltura sulle stesse superfici non può avvenire prima dei due anni.

Le porzioni di terreno devono quindi essere ben distinte tra loro e per ciascuna porzione deve essere mantenuto un registro degli interventi colturali.

La preparazione del terreno avviene con un' aratura profonda circa 40 cm, (se è disponibile viene interrato il letame), seguita da erpicature via via più leggere

La tecnica usuale di impianto è il trapianto, poiché la semina diretta con successivo dirado, sebbene migliore, risulta economicamente improponibile.

Non è permesso l'uso di varietà derivanti da organismi geneticamente modificati.

Per questo è necessario avere una certificazione della provenienza delle piantine o dei sementi data da etichette o cartellini forniti dal produttore

Durante la fase vegetativa il cavolo nero necessita di irrigazione e di interventi nutritivi.

Non è ammesso l'utilizzo di reflui provenienti da impianti di trasformazione e/o di depurazione.

Per quanto riguarda la fertilizzazione, le dosi massime apportabili di elementi nutritivi sono le seguenti:

| | | |
|----------|-----|-----------|
| Azoto | 80 | Kg/ettaro |
| Fosforo | 60 | Kg/ettaro |
| Potassio | 140 | Kg/ettaro |

Magnesio 25 Kg/ettaro

L'apporto di Azoto deve essere frazionato in almeno due interventi (a meno che l'apporto totale non sia inferiore o uguale al 50% dell'apporto massimo consentito)

Nel caso di fertirrigazione le quantità massime vanno ridotte del 20%.

È obbligatorio un piano di fertilizzazione redatto da un tecnico con titolo di studio nel settore agricolo. Per la verifica del rispetto di tale piano è necessario un registro di magazzino in cui vengono registrate le scorte di fertilizzanti. Gli interventi vanno registrati nel registro degli interventi colturali.

La raccolta viene effettuata quando la “testa” ha raggiunto le dimensioni tipiche della cultivar e risulta ben formata e di aspetto compatto.

Il numero e la frequenza delle raccolte dipende dal tipo di cultivar ma di solito si effettuano dalle 2 alle 3 raccolte.

La raccolta consiste nel recidere la pianta a livello del terreno e privandola delle foglie basali con un taglio netto.

È obbligatorio pulire il prodotto da ogni tipo di residuo proveniente dal campo.

Per il trasporto alla fase di condizionamento, il prodotto alla raccolta deve essere collocato in contenitori di plastica lavabili o contenitori monouso in legno o cartone.

Il prodotto deve essere lavorato e condizionato entro le 6 ore dalla raccolta.

Per garantire l'intera filiera produttiva è necessario che il prodotto sia mantenuto in aree separate e distinte da produzioni con diversa origine.

Le partite saranno contrassegnate da cartellini che indicheranno nome del produttore, varietà e data di raccolta.

Il prodotto dovrà essere lavorato su linee separate o, dove non possibile, sulla stessa linea ma in tempi diversi e solo dopo che questa è stata sanificata.

Si deve sempre garantire la separazione spazio temporale tra prodotti di diversa origine.

Le caratteristiche che il prodotto deve presentare dopo condizionamento e imballaggio sono:

- non aperti o prefioriti
- sani, non affetti da marciume
- non ammaccati o altrimenti danneggiati
- esenti da insetti o parassiti
- esenti da alterazioni causate da gelo
- puliti, privi di sostanze estranee visibili

Il prodotto deve essere suddiviso per categorie (determinate dall'aspetto).

Il prodotto deve essere diviso per calibrazione (determinata dal peso netto) cioè il peso della palla più pesante non deve superare il doppio di quello della palla più leggera nello stesso imballaggio.

Il peso netto unitario non deve essere inferiore ai 350 grammi (sono previste tolleranze del 10% in numero o in peso)

Ogni imballaggio (o ogni partita) deve contenere soltanto prodotti provenienti dalla stessa origine ed essere privi di qualsiasi corpo estraneo.

La parte visibile dell'imballaggio o della partita deve essere rappresentativa dell'insieme.

I materiali usati per l'imballaggio devono essere nuovi, puliti e di sostanze che non possono provocare alterazioni interne o esterne al prodotto.

Ogni imballaggio deve recare in caratteri allineati sullo stesso lato, leggibili, indelebili e visibili dall'esterno le seguenti indicazioni:

- identificazione imballatore e/o speditore
- natura del prodotto se non visibile dall'esterno
- origine del prodotto
- caratteristiche commerciali (categoria,peso.....)

Il prodotto può essere conservato in apposite celle frigorifere ma non possono trascorrere più di 5 giorni dalla raccolta alla vendita al consumatore

Il trasporto può avvenire solo con mezzi dotati di cella frigorifera o coibentati.

Questi rappresentano i processi essenziali e le regole da seguire per la produzione del cavolo nero.

Risultano quindi individuati gli attori responsabili, cioè coloro che gestiscono i prodotti primari (ricordiamo che ai fini della tracciabilità non è necessario tenere traccia di elementi secondari quali ad esempio additivi chimici, imballaggi, ecc... ma per questi basta semplicemente registrare i fornitori)

Le aziende della filiera risultano quindi:

- **Fornitore di piante**

Questa è l'azienda che si occupa della coltivazione in serra delle sementi per ottenere le piantine che verranno vendute ai coltivatori.

Si occupa inoltre della coltivazione in campo aperto di alcune delle piantine stesse al fine di riottenere nuove sementi.

- **Coltivatore**

È l'azienda che si occupa della coltivazione, della raccolta e di una prima selezione e stoccaggio dei cavoli.

- **Confezionatore.**

Questa è l'azienda che si occupa del condizionamento e del confezionamento dei cavoli raccolti.

- **Distributore**

È l'azienda che si occupa della distribuzione e della vendita al dettaglio dei cavoli confezionati.

È possibile estrarre uno schema riassuntivo in cui sono specificati gli attori principali, le loro attività principali ed i siti in cui queste attività vengono svolte.

| FORNITORE PIANTE | COLTIVATORE DIRETTO | CONFEZIONATORE | DISTRIBUTORE |
|---|--|---|--|
| <p>processi principali:</p> <ul style="list-style-type: none"> - coltivazione in campo aperto (per sementi) - coltivazione in serra - vendita <p>caratteristiche</p> <ul style="list-style-type: none"> - data di semina <p>siti</p> <ul style="list-style-type: none"> - campo aperto - serra - magazzino sementi | <p>processi principali</p> <ul style="list-style-type: none"> - coltivazione in campo aperto - raccolta e imballaggio - deposito - acquisto - vendita <p>caratteristiche:</p> <ul style="list-style-type: none"> - data di raccolta - peso della cassa - consistenza (chiuso, semi-aperto) - azoto, fosforo, potassio, magnesio (sul terreno) - danneggiamento (basso, medio, alto) <p>siti:</p> <ul style="list-style-type: none"> - campi - magazzino di raccolta - magazzino sementi (concimi) | <p>processi principali:</p> <ul style="list-style-type: none"> - condizionamento - classificazione (in base a peso e danneggiamento) - imballaggio - conservazione - etichettatura <p>caratteristiche</p> <ul style="list-style-type: none"> - categoria - data di lavorazione - linea (processo) - temperatura (conservazione) <p>siti:</p> <ul style="list-style-type: none"> - linea di lavorazione - magazzini di raccolta pre-lavorazione - magazzini di conservazione | <p>processi principali:</p> <ul style="list-style-type: none"> - acquisto - deposito - vendita all'ingrosso <p>caratteristiche:</p> <ul style="list-style-type: none"> - temperatura di conservazione - data di cessione <p>siti:</p> <ul style="list-style-type: none"> - magazzino di deposito |
| | compravendita di piante | compravendita di casse di cavolo | compravendita di imballi di cavolo |

2.3 Introduzione a UML

UML è uno standard di modellazione visuale che consente la rappresentazione, tramite diagrammi, di qualunque tipo di sistema, sia esso hardware, software, organizzativo... Nacque nella seconda metà degli anni novanta dallo sforzo congiunto di tre ricercatori della Rational Software, Rumbaugh, Jacobson e Booch, come fusione ed estensione di altri formalismi e linguaggi preesistenti. Attraverso aggiunte e modifiche, UML è attualmente giunto alla versione 2.0 e si è imposto come standard de facto per la progettazione.

UML propone diversi diagrammi per rappresentare un sistema, i quali appartengono principalmente a due categorie: i diagrammi “strutturali” ed i diagrammi “comportamentali”. Fra queste categorie verranno utilizzati i diagrammi delle classi per quanto riguarda gli strutturali, mentre useremo i diagrammi di attività e quelli di sequenza per quanto riguarda quelli comportamentali.

Diagramma delle classi.

Una classe rappresenta astrattamente un insieme di entità che hanno le stesse caratteristiche, cioè gli stessi attributi che li identificano e stesse operazioni, relazioni e comportamenti.

Una classe può rappresentare oggetti reali, ad esempio dei macchinari, oppure degli oggetti astratti, come ad esempio una transazione bancaria.

La forma minima per rappresentare una classe consiste in un rettangolo contenente solo il nome della classe.

Una espansione di questa rappresentazione è formata da un box rettangolare diviso in tre parti dove la prima contiene il nome della classe, la seconda parte contiene gli attributi della classe e la terza le operazioni.

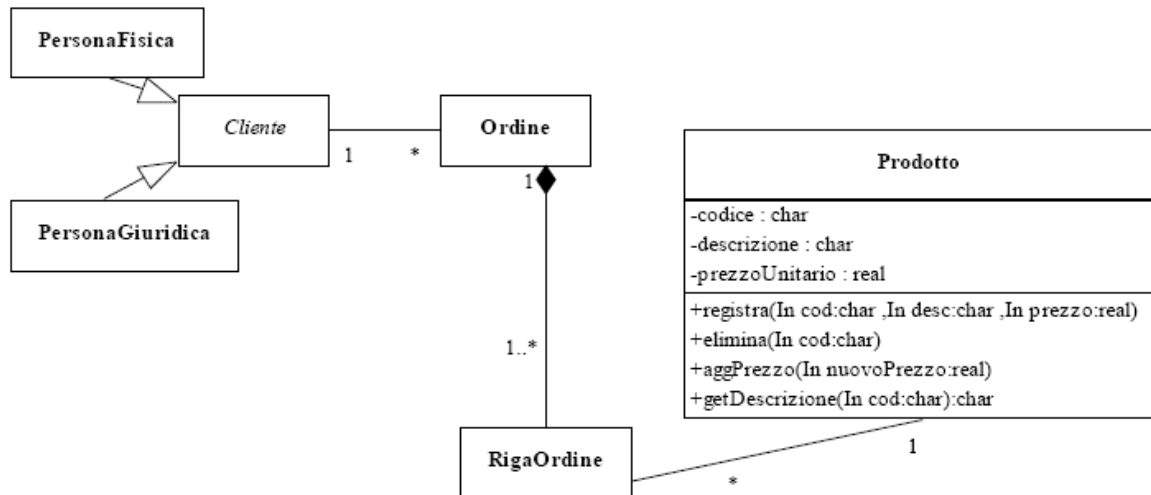


Figura 5 Esempio di diagramma delle classi

Un oggetto, che è un'istanza della classe, viene rappresentato come una classe, ma al posto del nome ci va il nome dell'istanza preceduto dal nome della classe divisi da : e sottolineati. Inoltre, eventualmente, la lista degli attributi con i loro valori attuali.

La rappresentazione esplicita di oggetti in UML di solito è rara, essendo una notazione di specifica e non implementativa, comunque può risultare utile per esemplificare delle situazioni particolari o tipiche.

Le classi vengono relazionate tra loro attraverso associazioni, aggregazioni e composizioni

Un'associazione è una relazione semplice che lega due classi ed è rappresentata da una linea che le collega. Alle estremità è possibile inserire la cardinalità dell'associazione ed è inoltre possibile inserire un nome caratteristico per l'associazione stessa.

In UML sono disponibili anche altri due tipi di associazioni: l'aggregazione e la composizione

L'aggregazione viene rappresentata da una linea con un rombo vuoto (dalla parte dell'entità complessa) ad un'estremità e rappresenta una relazione che lega un'entità complessa alle sue parti componenti, dove queste possono appartenere anche ad altri oggetti.

La composizione invece è una aggregazione in cui si richiede la partecipazione esclusiva del componente all'oggetto composto e si rappresenta con un rombo pieno sulla linea dalla parte dell'entità composta.

Un'altra relazione che può collegare due classi è quella di generalizzazione. La generalizzazione introduce una gerarchia di classi e viene rappresentata da una linea con una freccia vuota dalla parte della classe padre.

La classe figlio eredita le caratteristiche della classe padre ed un'istanza della derivata è anche un'istanza della classe padre.

Diagrammi di sequenza

Un diagramma di sequenza mostra lo scambio di messaggi fra tutti gli oggetti che vengono via via coinvolti in una attività iniziata da un attore. In altre parole, un diagramma di sequenza espande e specifica un caso d'uso.

Il diagramma si sviluppa dall'alto verso il basso seguendo un'ipotetica linea temporale. Gli oggetti (istanze di classi) che partecipano all'esecuzione vengono tutti indicati nella parte alta del diagramma con un rettangolo che ne riporta il nome e/o il tipo (la notazione e' la stessa dei diagrammi delle classi). La vita di ogni oggetto viene rappresentata da una linea tratteggiata verticale: su di essa possono essere tracciati stretti rettangoli più o meno alti, ad indicare i periodi di entrata in attività dell'oggetto.

Gli oggetti possono essere creati o distrutti da altri oggetti in corso di esecuzione. Un oggetto ne può creare un altro attraverso uno speciale messaggio di creazione: diversamente dagli altri, posizionati nella parte alta del diagramma, il nuovo oggetto viene inserito in mezzo al diagramma, all'altezza del messaggio che lo ha generato. La distruzione di un oggetto viene invece indicata da una grande X sulla linea di vita: un oggetto può deallocarsi autonomamente, oppure essere distrutto tramite un messaggio.

Lo scambio dei messaggi e' indicato con delle frecce fra le linee di vita di due oggetti. Le frecce sono generalmente marcate col contenuto (logico) del messaggio, ma prima di questo possono essere presenti delle ulteriori diciture.

Una scrittura fra parentesi quadre indica un test condizionale il cui risultato determina l'invio del messaggio. Un'istruzione di assegnazione specifica che il risultato del messaggio diventa il valore di una certa variabile. Infine, un asterisco (*) segnala che l'operazione di invio deve essere iterata.

Le frecce hanno significati diversi in base al modo in cui sono disegnate.

Le frecce dirette a punta piena indicano messaggi sincroni, mentre quelle tratteggiate sono risposte di ritorno. Quando hanno solo mezza punta rappresentano invece i messaggi asincroni. Sono possibili anche frecce che ritornano nell'oggetto da cui partono, per indicare chiamate a funzioni interne:

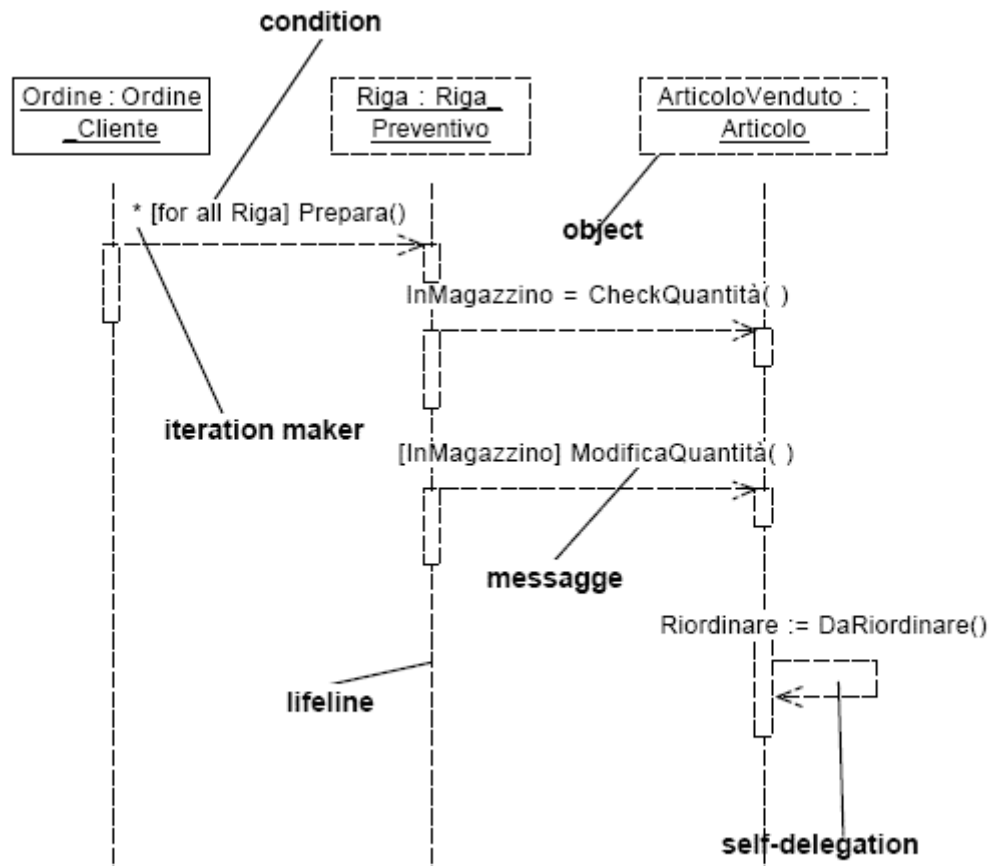


Figura 6 Esempio di diagramma di sequenza

Diagrammi di attività

Il diagramma di attività descrive la sequenza delle attività e supporta un comportamento condizionale e parallelo. Si tratta di una variante dei diagrammi di stato in cui tutti gli stati hanno associata una attività.

Gli elementi di un diagramma di attività sono:

- **Attività:** indicano un lavoro che deve essere svolto. Da ogni activity possono uscire uno o più **transazioni**, che indicano il percorso da una activity ad un'altra.
- **Start e End Point:** punti di inizio e fine del diagramma. Gli End Point possono anche non essere presenti, oppure essere più di uno.
- **Branch e merge :** il comportamento condizionale è determinato da questi due costrutti. Il branch ha una singola transazione entrante e più transazioni uscenti in cui solo una di queste sarà prescelta. Il merge ha più transazioni entranti e una sola uscente e serve a terminare il blocco condizionale cominciato con un branch.
- **Fork e join:** il comportamento parallelo è determinato da questi due costrutti. Quando scatta la transazione entrante, si eseguono in parallelo tutte le transazioni che escono dal fork. Con il parallelismo

non è specificata la sequenza. Per la sincronizzazione delle attività parallele è presente il costrutto di join che ha più transazioni entranti ed una sola transazione uscente.

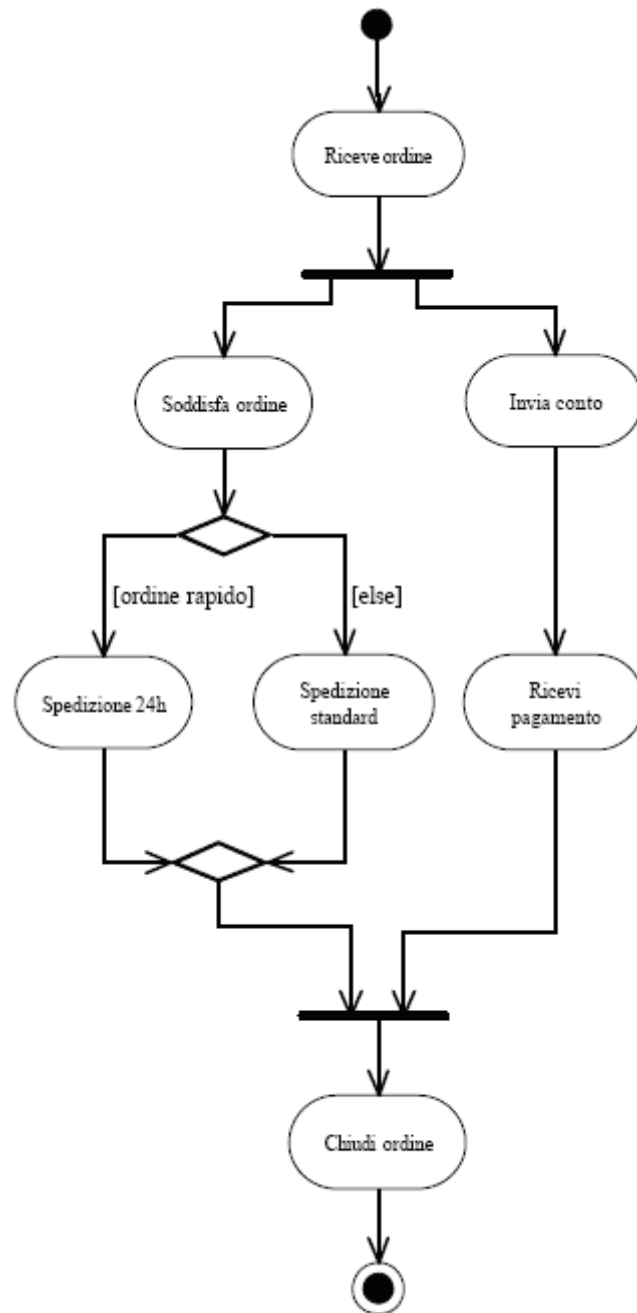


Figura 7 Esempio di diagramma di Attività

2.4 Introduzione al modello Cerere

Una volta estratti le attività e gli oggetti principali da tracciare della filiera, questi devono essere tradotti in un modello informatico.

Noi prenderemo come riferimento il modello usato in CERERE, un sistema informatico per il supporto della tracciabilità nelle filiere agroalimentari.

Gli oggetti che questo sistema supporta sono espressi, con i loro principali attributi, nel diagramma seguente:

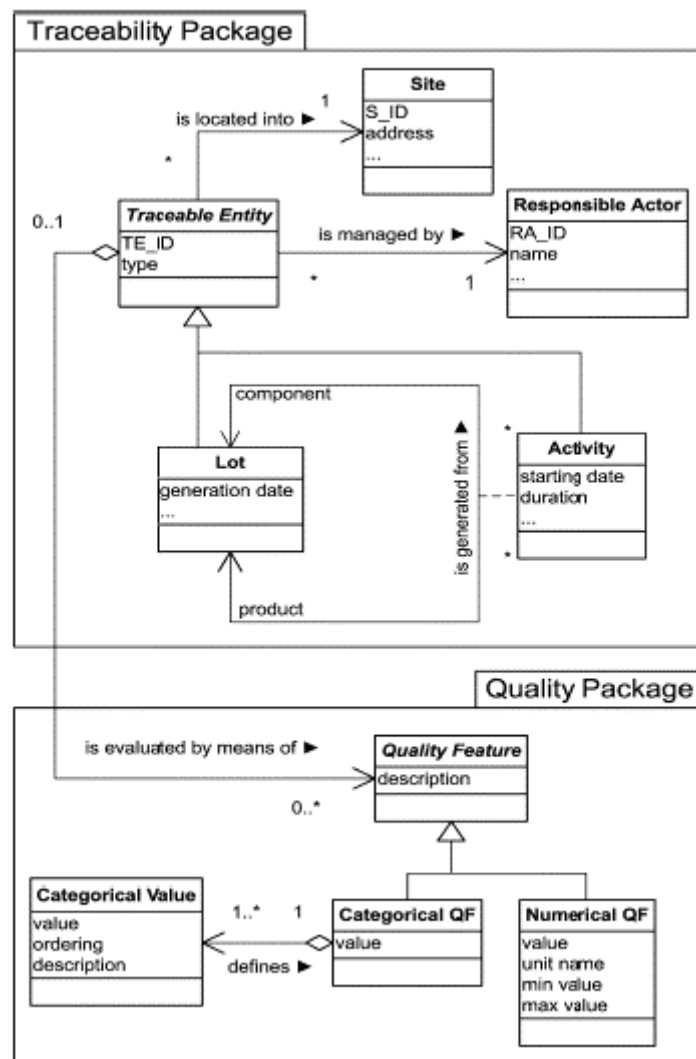


Figura 8 Diagramma delle classi del modello per la tracciabilità

Come possiamo vedere la classe principale è l'Entità *Tracciabile* dalla quale derivano la classe *Lotto* e la classe *Attività*, anch'essa vista come entità tracciabile. A questa classe possono essere relazionate alcune caratteristiche di qualità, che possono specializzarsi in caratteristiche di tipo numerico, come ad esempio il peso, o in caratteristiche di tipo categorico, come ad esempio la consistenza.

L'entità tracciabile è utilizzata da un attore responsabile ed è sempre associata ad un preciso sito

La visione dinamica del sistema è data dai seguenti diagrammi:

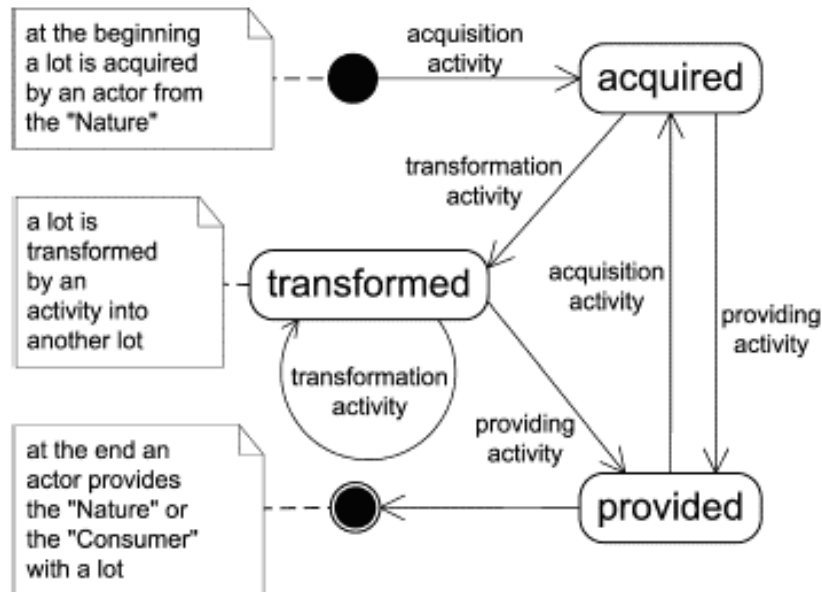


Figura 9 Diagramma di stato per l'entità lotto

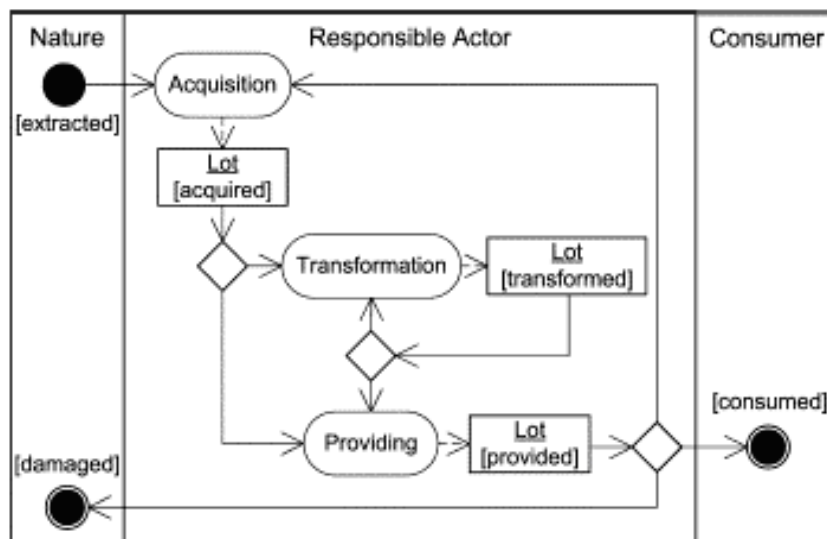


Figura 10 Diagramma di attività per un lotto

All'inizio c'è, da parte di un *Attore Responsabile*, l'acquisizione di materie prime dalla natura, quindi il lotto passa allo stato *acquisito*.

Da qui è possibile trasformare il lotto, e quindi spostarsi nello stato *trasformato*, o vendere il lotto stesso, che quindi passa nello stato *ceduto*.

Dallo stato trasformato, il lotto può subire ulteriori trasformazioni, e quindi rimanere nello stato trasformato, oppure essere ceduto.

Ciò può accadere per vari motivi, tra i quali lo scarto (di nuovo verso la natura, ma questa volta come punto di arrivo), la cessione ad un altro attore della filiera, oppure per la cessione al consumatore finale (che rappresenta un altro end point) .

Una volta ceduto il lotto non è più sotto la responsabilità di quel determinato attore della filiera, e questa ricadrà su di un nuovo attore se questo esegue una acquisizione, o sulla natura o sul consumatore se finisce il ciclo.

La figura seguente mostra i diagrammi di sequenza per il prototipo i quali mettono in luce l'esistenza di un database centralizzato che registra tutte le informazioni relative ai lotti, compresi i cambi di responsabilità tra un attore e l'altro e le movimentazioni e le trasformazioni dei lotti.

È necessario quindi che queste gli siano inviate direttamente dagli attori coinvolti nelle varie operazioni.

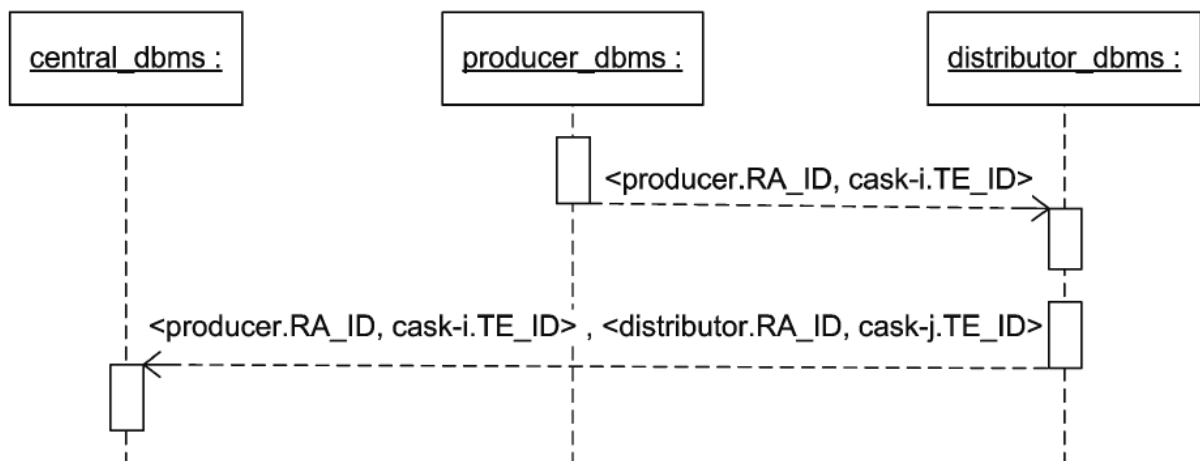


Figura 11 Diagrammi di sequenza per il prototipo cerere

2.5 Modelli dinamici per la filiera del cavolo nero

Come si evince dal diagramma di attività in figura 12, il fornitore delle piantine ottiene le sementi di partenza dalla natura, queste verranno raggruppate in balle e conservate in appositi magazzini, oppure utilizzate per la coltivazione in campo aperto, con cui ottenere nuove sementi.

Quando un Coltivatore necessita di nuove piantine, il fornitore preleva le sementi dal magazzino e, tramite l'attività di coltivazione in serra, trasforma le sementi ottenendo delle casse alveolari contenenti piantine di cavolo.

A questo punto può avvenire la cessione del lotto con la consegna al coltivatore attraverso il proprio furgone.

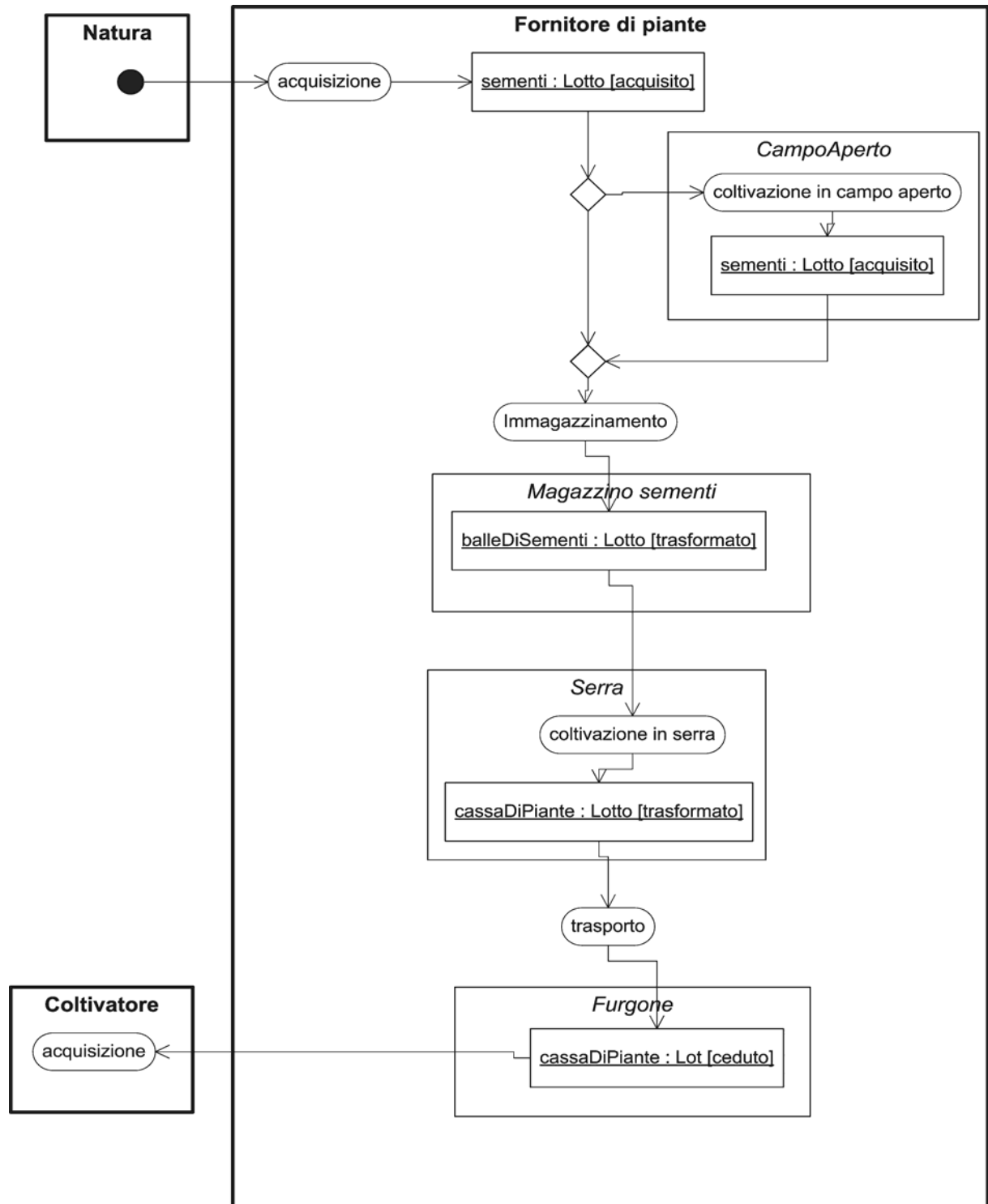


Figura 12 Diagramma di attività Fornitore Pianta

Il coltivatore, come si può vedere in figura 13, dopo aver acquistato la cassa (o le casse) di piante dal fornitore (e di conseguenza averne preso la responsabilità), inizia l'attività di coltivazione in campo aperto, che rappresenta la vera e propria produzione del cavolo. Al termine dell'attività di coltivazione, sempre compresa in essa, c'è la raccolta dei cavoli pronti, la quale produce delle casse di cavoli.

Queste casse di cavioli vengono portate in appositi magazzini in attesa di essere ritirate dal furgone del confezionatore che nel frattempo le ha acquisite.

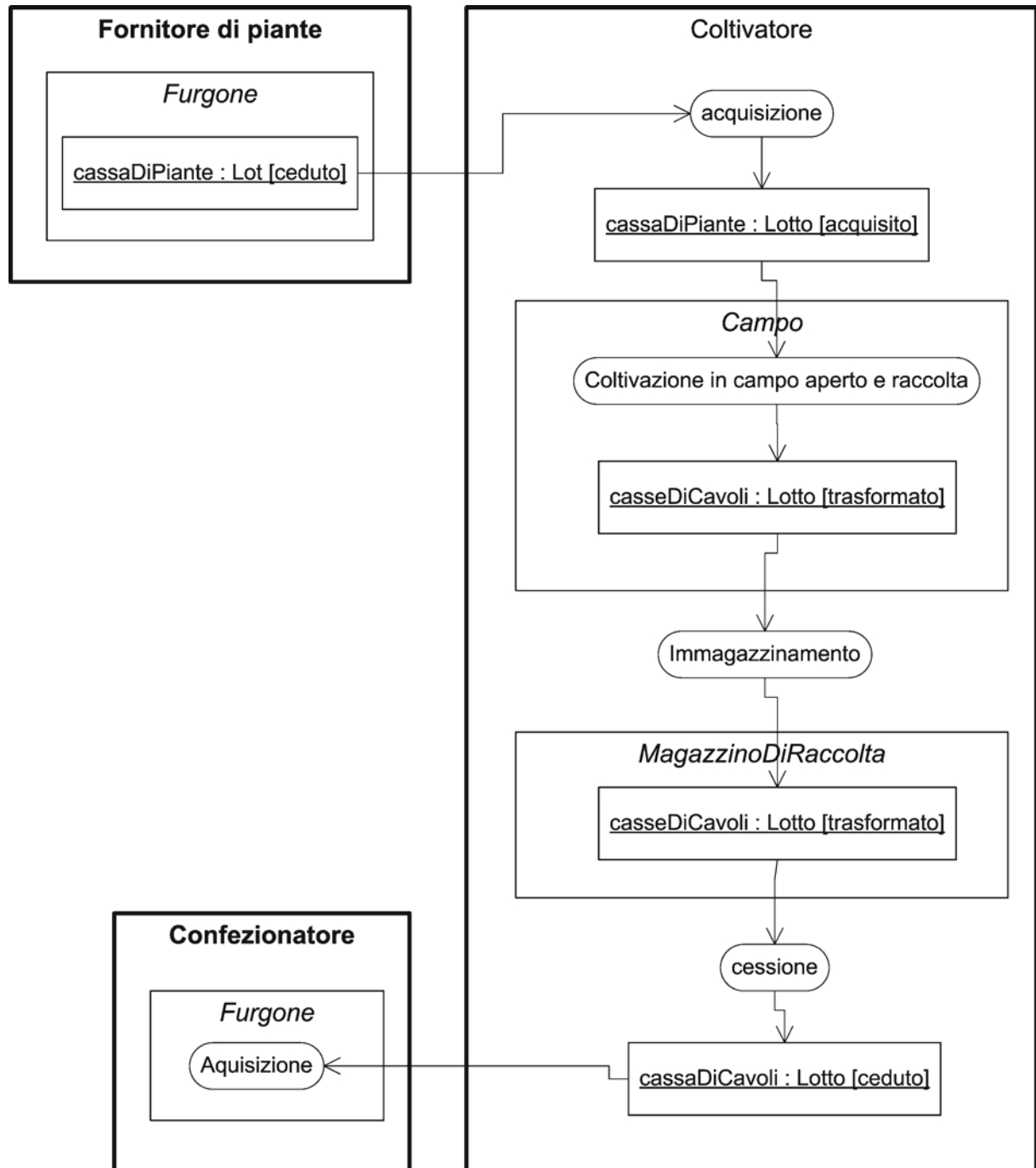


Figura 13 Diagramma di attività Coltivatore

Il confezionatore acquista delle casse di cavioli dal coltivatore e le va a prendere col proprio furgone, dopodichè le porta presso le proprie linee di lavorazione.

Qui, i cavioli subiscono la prima lavorazione che consiste nel condizionamento, cioè lo scarto dei pezzi inadeguati, i quali vengono nuovamente ceduti alla natura, la pulizia da tutti i corpi estranei provenienti dal terreno e l'eliminazione delle foglie esterne che presentano danneggiamenti dovuti a gelo o alla manipolazione.

Dopo l'attività di condizionamento avviene la classificazione che produce una selezione in base alle caratteristiche del prodotto in modo da ottenere due flussi materiali, entrambi i quali proseguono, distintamente gli uni dagli altri .

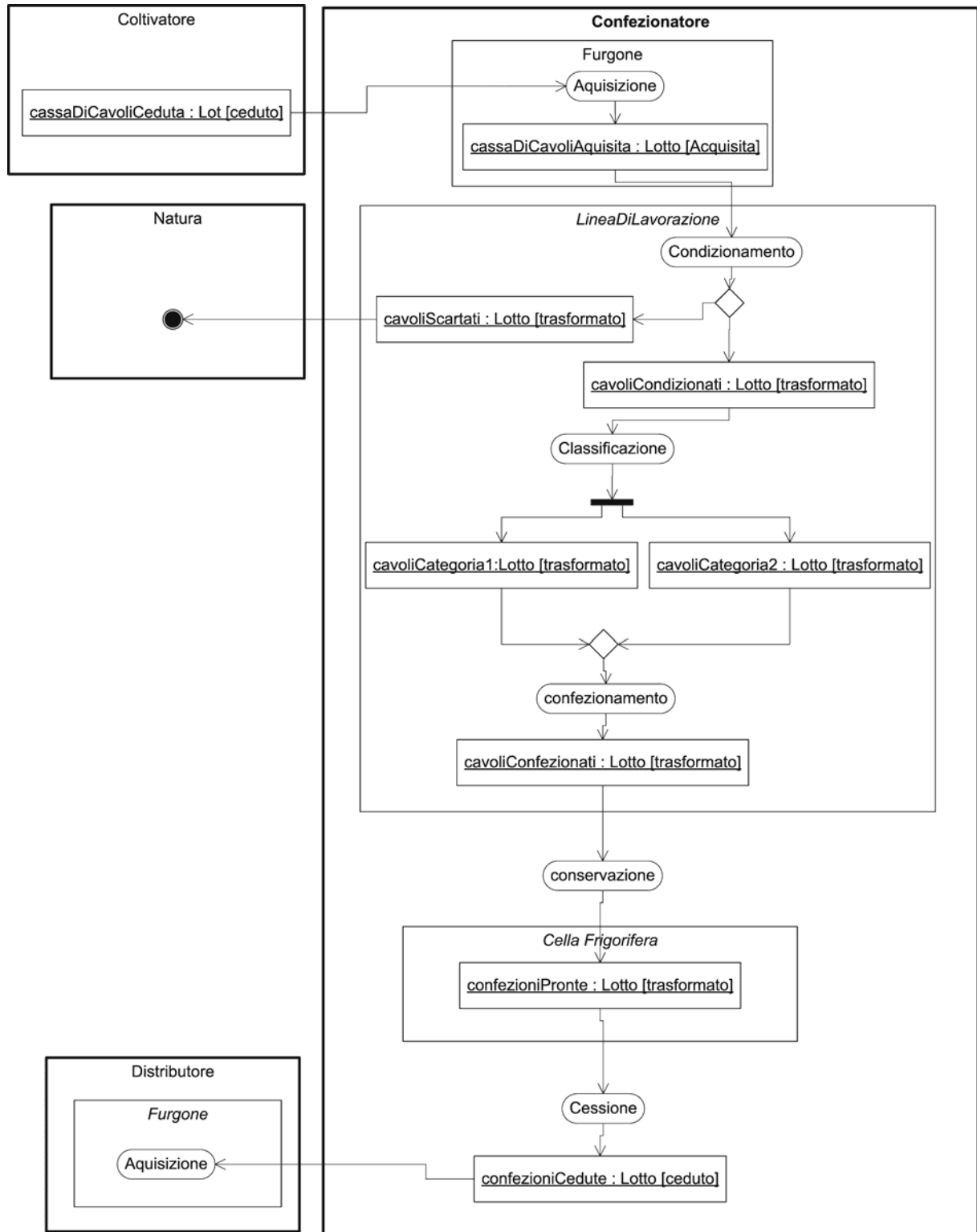


Figura 14 Diagramma di attività Confezionatore

È da notare che la separazione dei flussi avviene in contemporanea (quindi è stata usata una fork nel diagramma) mentre dopo i flussi di prodotto viaggiano separatamente, quindi in maniera ben distinta l'uno dall'altro, verso il confezionamento (per questo nel modello è stato utilizzato un merge piuttosto che un join).

Come già detto i lotti viaggiano verso il confezionamento vero e proprio, ottenendo così in uscita dei cavi confezionati, i quali vengono immagazzinati e conservati in celle frigorifere nell'attesa di essere venduti ad un distributore. Una volta venduti è il distributore che si occupa con un proprio mezzo di ritirare le confezioni di prodotto.

Il distributore si procura da un confezionatore dei cavi condizionati e confezionati da immettere sul mercato, li va a ritirare presso l'azienda tramite il proprio furgone coibentato e li trasporta presso i propri siti dove avviene l'immagazzinamento in celle frigorifere in attesa della vendita al consumatore finale.

Per semplicità è stato preso in esame in questa tesi un solo passaggio nella catena di distribuzione anche se nella realtà, prima di arrivare realmente al consumatore finale, ci possono essere più passaggi, anche fra aziende diverse.

Questo non è molto influente ai fini della tracciabilità (basta infatti aggiungere nella filiera eventuali altri attori e/o siti) poiché la struttura sarà simile a quella da noi proposta.

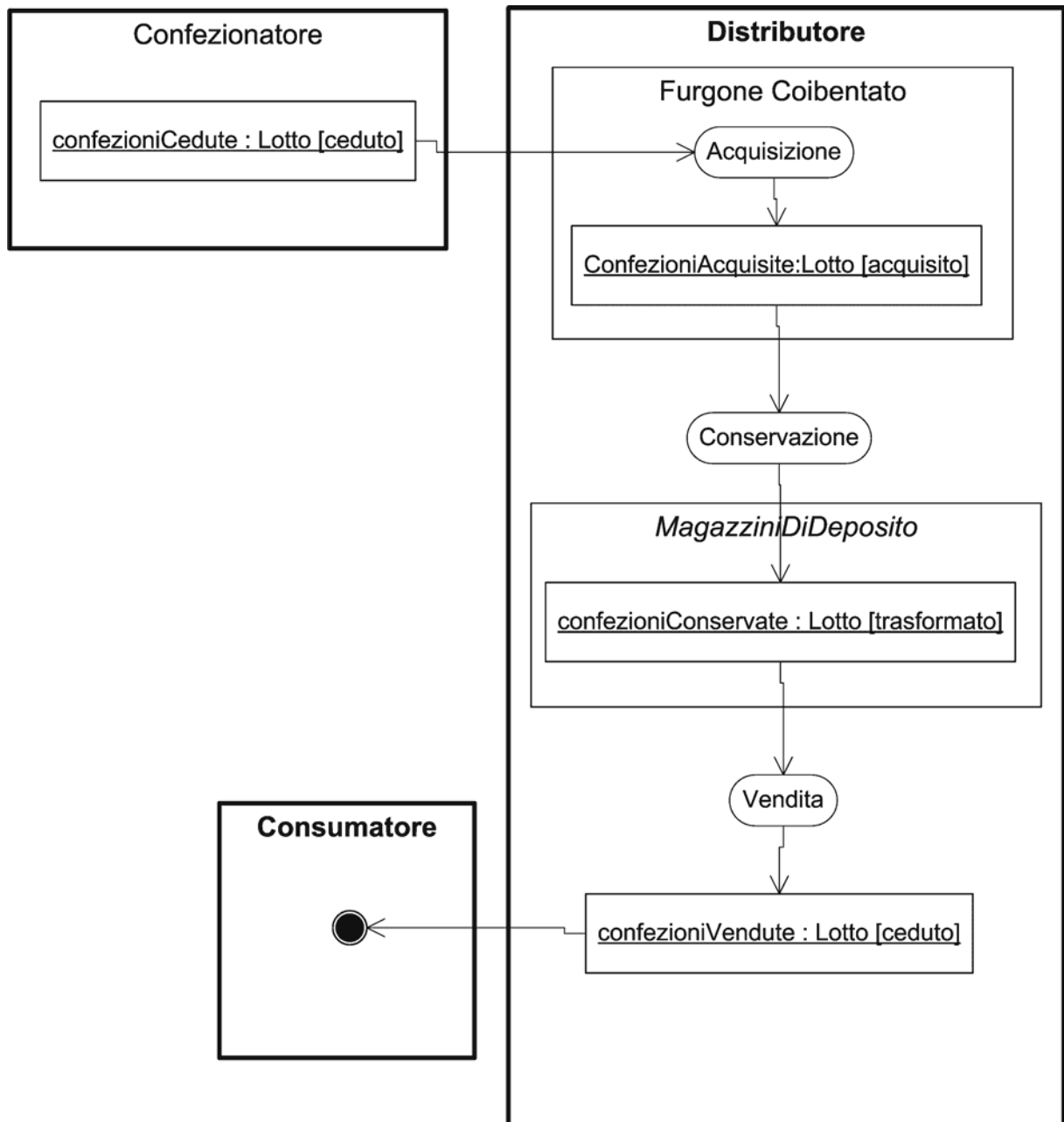


Figura 15 Diagramma di attività Distributore

2.6 Modelli statici per la filiera del cavolo nero

In questo capitolo prenderemo in esame tutti gli oggetti (statici) che compongono il modello di filiera da noi tracciata, basandoci sulle classi *Attività*, *Lotto*, *Attore Responsabile*, *Sito*, secondo il modello delle classi usato in Cerere.

L'elemento di partenza sono le sementi che vengono estratte dalla natura o ottenute dall'attività di coltivazione in campo aperto; la caratteristica di questo oggetto, oltre a quelle standard per tutti gli oggetti di tipo *Lotto*, che sono *ID* e *dataDiGenerazione*, è rappresentata dalla caratteristica numerica *Peso*, espresso in KG, che esprime il peso delle sementi estratte o raccolte. L'attore responsabile ed il sito associati al lotto non presentano caratteristiche salienti ai fini della tracciabilità oltre a quelle normalmente presenti negli oggetti di tipo *AttoreResponsabile* e *Sito*

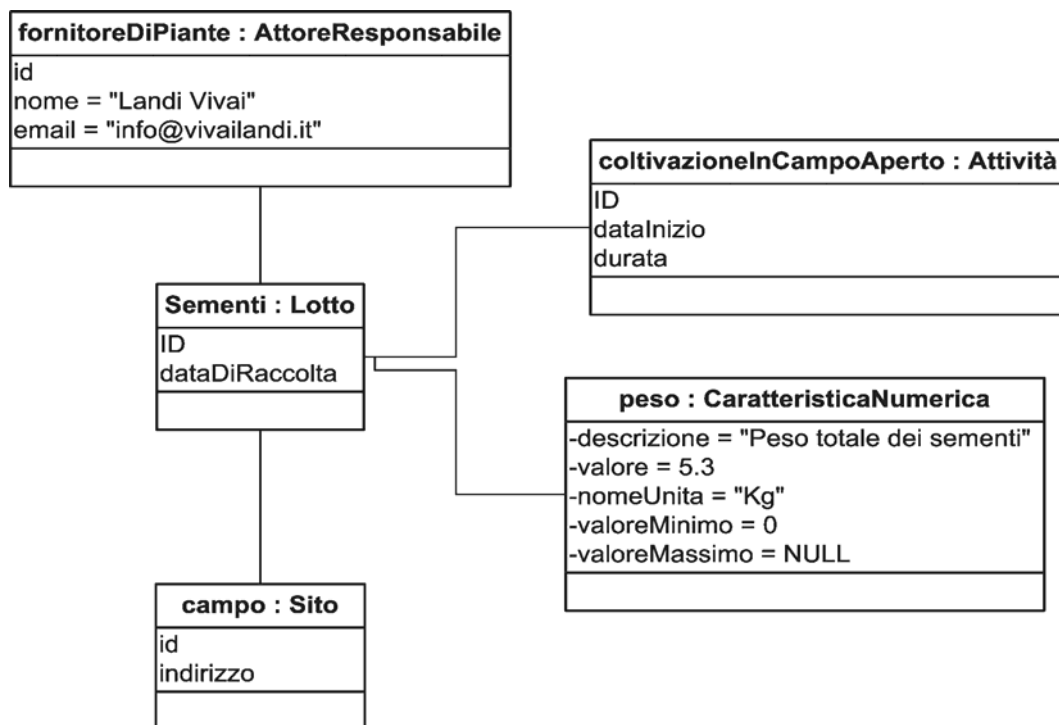


Figura 16 Struttura statica per Sementi

Una volta ottenute le sementi queste devono essere immagazzinate. Per fare questo le sementi vengono raggruppate o divise in balle, più comode per lo stoccaggio in magazzino.

Anche in questo caso l'unica caratteristica saliente da aggiungere a quelle normalmente presente tra gli oggetti è il *peso* della balla di sementi, caratteristica di tipo numerico, espressa in Kg.

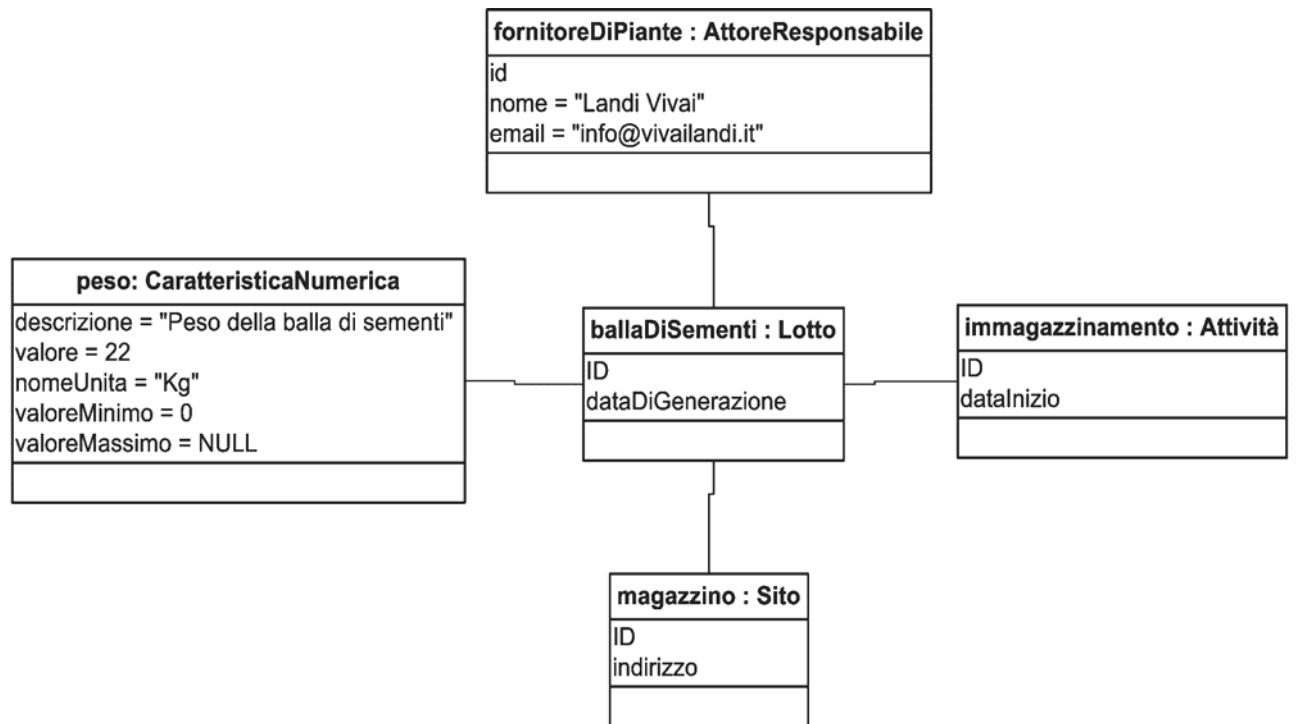


Figura 17 Struttura statica Balla di sementi

L'attività di *coltivazione in serra* produce l'oggetto *cassaDiPiante* che rappresenta un contenitore alveolare in cui vengono seminati i cavoli che è poi il prodotto principale di questo attore.

Tralasciando tutte le fasi che riguardano la coltivazione in serra, che rappresentano più dei processi interni che dei processi importanti ai fini della tracciabilità, quello che ci interessa fondamentalmente è, oltre alla *data di semina*, il *numero delle piantine* effettivamente presenti all'interno dei contenitori.

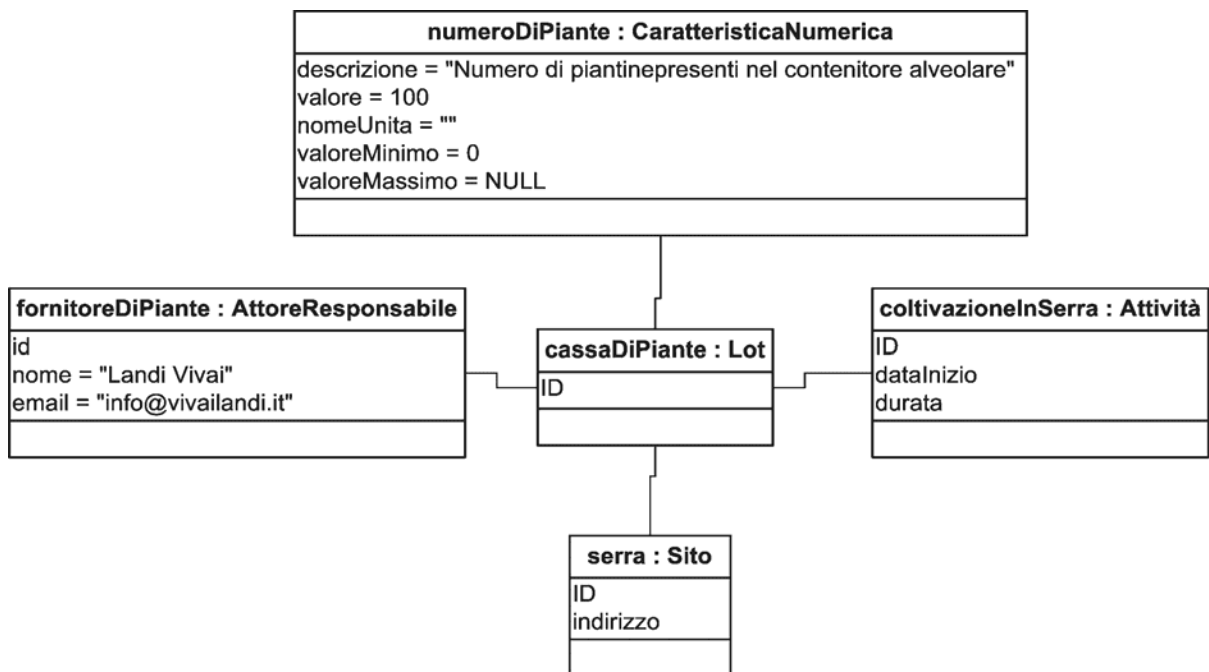


Figura 18 Struttura statica per cassa di piante

In questo caso c'è solo da registrare l'attività *cessione* che avviene nel sito *furgone* (attività che implica quindi anche il trasporto) e che comporta il cambio di attore responsabile. Nuovamente è presente la caratteristica numerica relativa al numero di piantine contenute nella cassa.

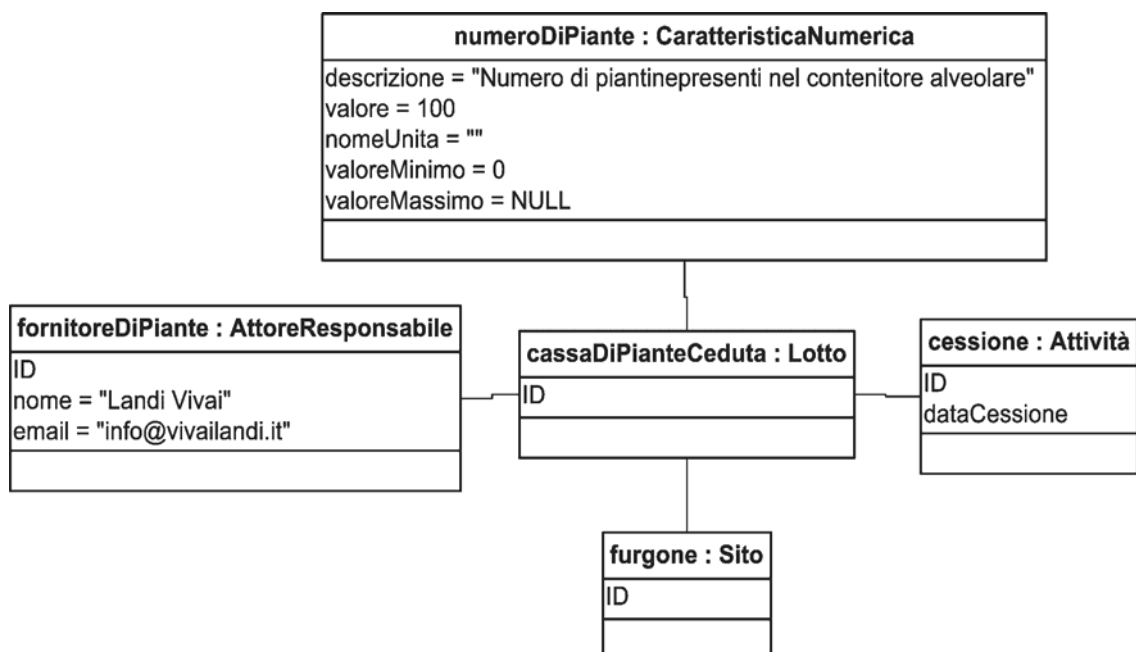


Figura 19 Struttura statica per cassa di piante ceduta

In seguito abbiamo la registrazione da parte del coltivatore dell'acquisizione di lotti di *casce di piante* provenienti dall'attore *fornitore*.

Non ci sono cambiamenti di stato negli oggetti ma è necessario registrare il cambiamento di responsabilità e di sito.

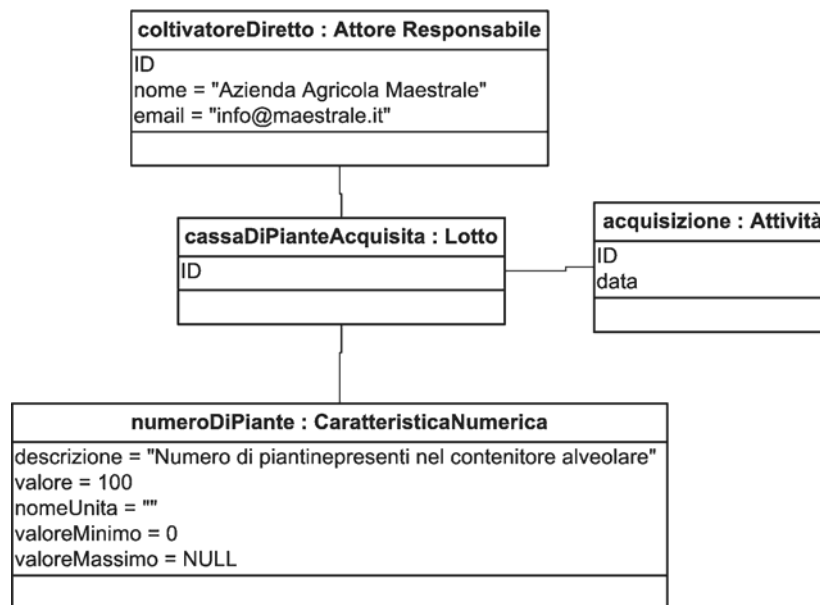


Figura 20 Struttura statica per cassa di piante acquisita

La parte seguente è la più importante per quanto riguarda la qualità del prodotto, infatti si tratta dell'attività di coltivazione vera e propria, attività descritta da diverse caratteristiche numeriche che rappresentano gli elementi "fertilizzanti" immessi nel terreno e che vanno ad influire sulle caratteristiche e sul gusto del prodotto. Queste sono tutte caratteristiche numeriche espresse in Kg per ettaro di superficie.

Il prodotto dell'attività di coltivazione sono *casse di cavoli* raccolti, che prevederanno un attributo semplice che identifica il *numero di casse* costituenti il lotto ed una serie di caratteristiche che vedremo più in dettaglio nella figura 22.

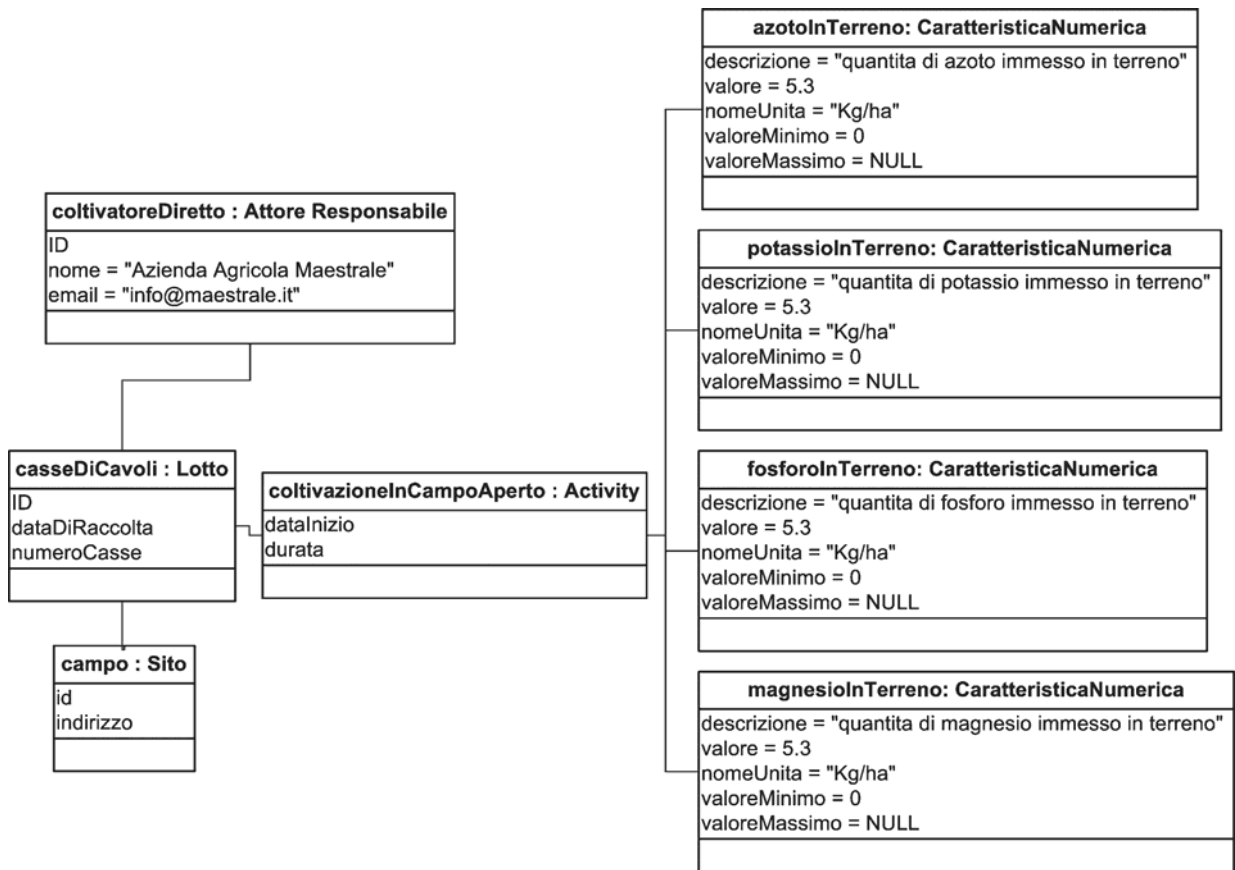


Figura 21 Struttura statica per casse di cavoli

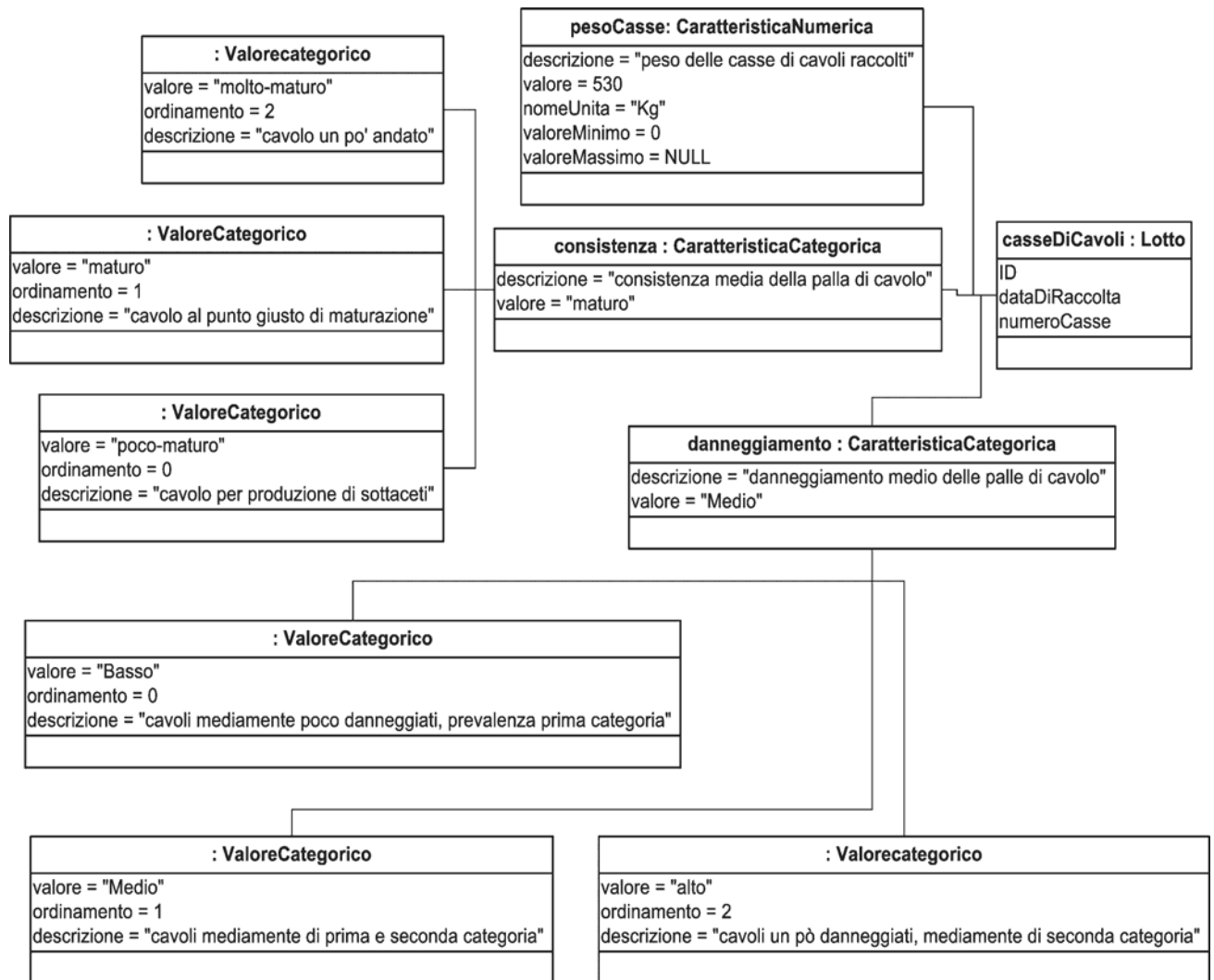


Figura 22 Casse di cavoli – caratteristiche lotto

Il lotto è caratterizzato da un valore numerico che rappresenta il peso totale del raccolto, oltre a caratteristiche categoriche come la *consistenza media* delle palle, cioè la compattezza del prodotto, ed il *danneggiamento medio* delle palle, entrambi indici della qualità del prodotto finale. Le cassette di cavoli, una volta riempite, vengono poste nei magazzini in attesa di essere cedute all'azienda addetta al confezionamento. Da notare che per la tracciabilità c'è bisogno di registrare lo spostamento al nuovo sito attraverso un nuovo lotto, anche se questo non ha cambiato le sue caratteristiche.

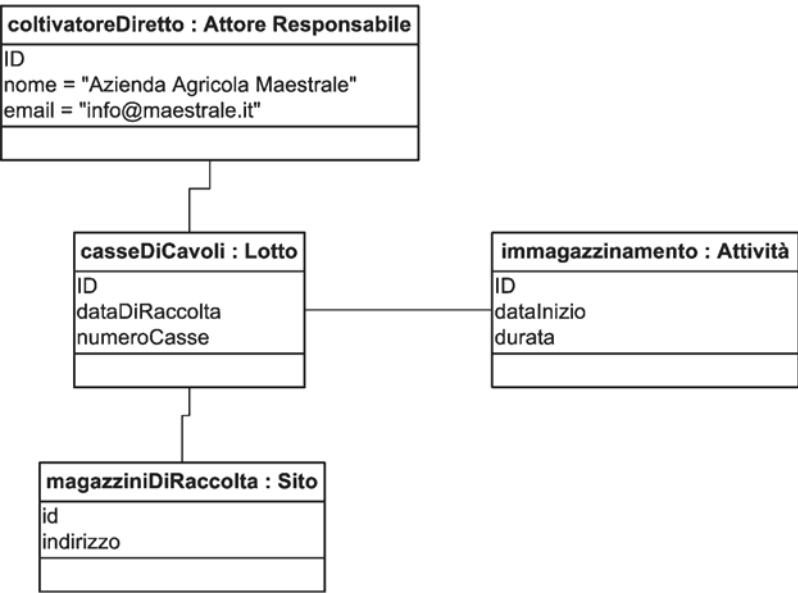


Figura 23 Struttura statica per casse di cavoli immagazzinate

Registrazione della cessione avvenuta all’azienda di confezionamento.

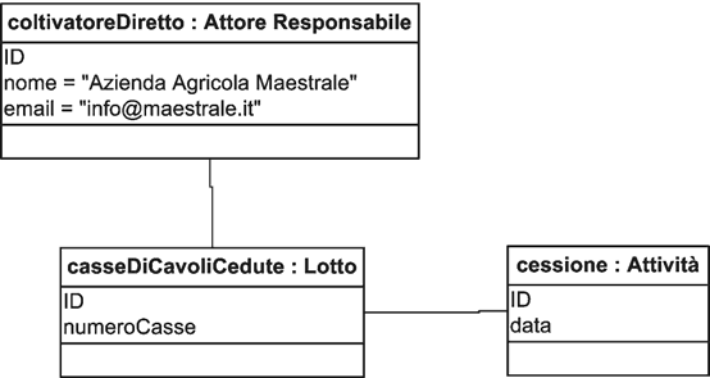


Figura 24 struttura statica per casse di cavoli cedute

Attraverso il prossimo stato si registra l’acquisto delle casse dei cavoli da parte del confezionatore ed il relativo cambio di responsabilità.

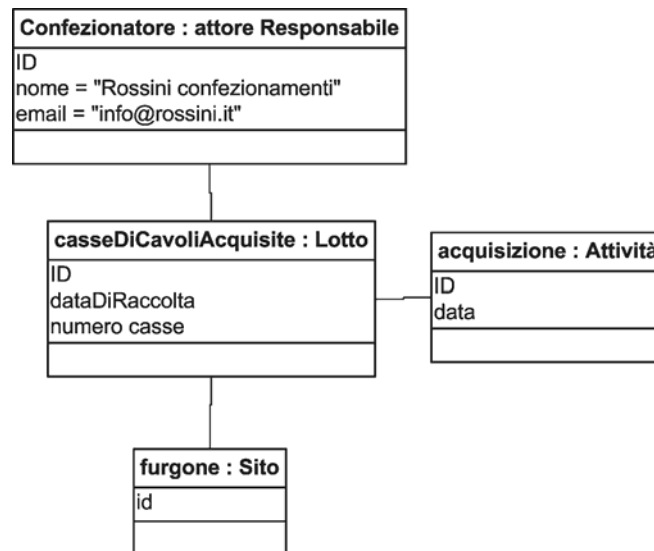


Figura 25 Struttura statica per casse di cavoli acquisite

La prima attività svolta dal confezionatore è il *condizionamento*.

Con condizionamento si intende la pulizia da tutti i corpi estranei presenti nei prodotti e la rimozione delle foglie esterne se danneggiate, inoltre prevede lo scarto alla natura dei prodotti troppo danneggiati. Viste queste operazioni il peso varia e quindi ritorna ad essere una caratteristica del prodotto dell'attività

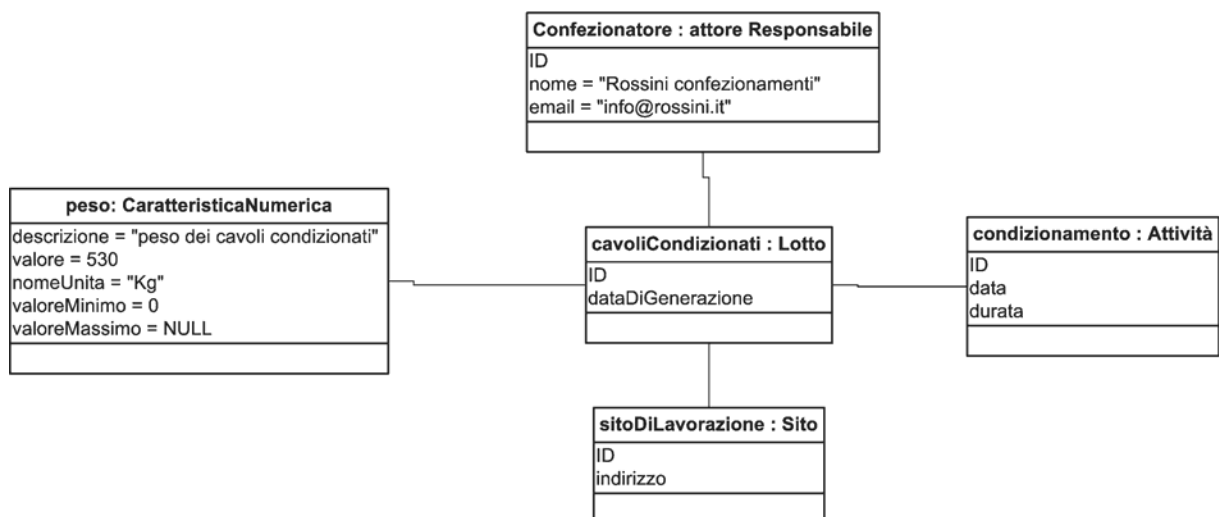


Figura 26 Struttura statica per cavoli condizionati

La *calibrazione* consiste nel dividere i cavoli in base alla *categoria* di appartenenza, data dal livello di danneggiamento esterno delle foglie della palla, quindi questa sarà una caratteristica importante per il prodotto dell'attività stessa.

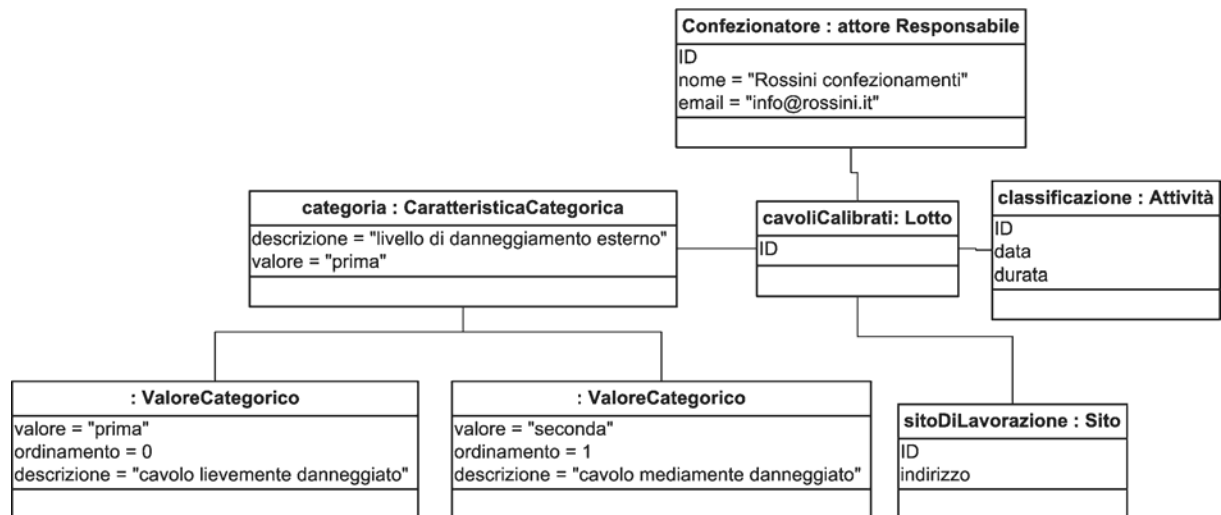


Figura 27 Struttura statica per cavoli calibrati

Una volta divisi per categoria i cavoli vanno confezionati.

Durante questa fase essi vengono generalmente divisi in lotti più piccoli.

Inoltre ogni lotto è generalmente composto da più di una confezione, per questo è necessario un campo che tenga traccia del *numero di confezioni* contenuto nel lotto.

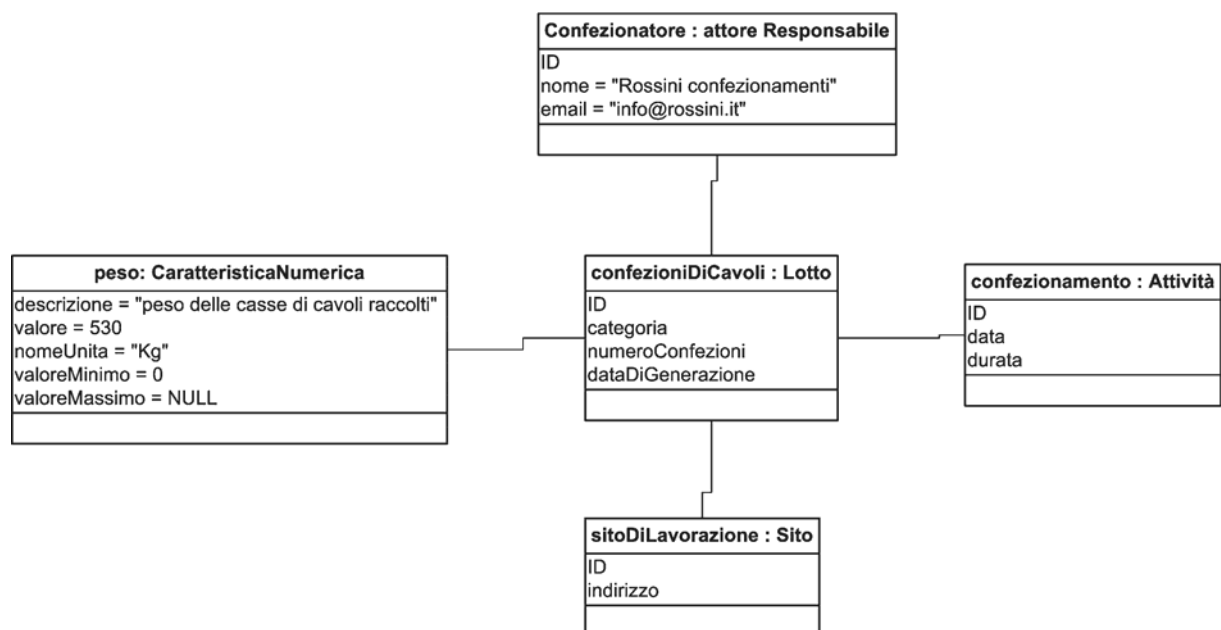


Figura 28 Struttura statica per confezioni di cavoli

Una volta che i lotti con le confezioni sono pronti, questi devono essere conservati in apposite celle frigorifere per evitare che il prodotto si deteriori in attesa di essere ceduto alle aziende distributrici. Una caratteristica importante sarà quindi rappresentata dalla temperatura di conservazione.

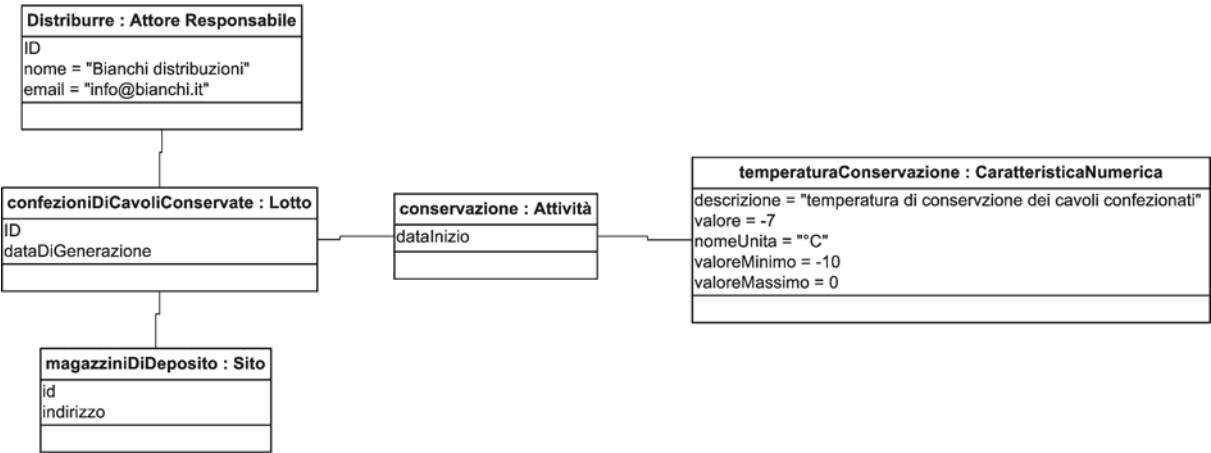


Figura 29 Struttura statica per confezioni di cavoli conservate

In Figura 30 si osserva lo stato per la registrazione della cessione delle confezioni di cavolo all’attore responsabile della distribuzione.

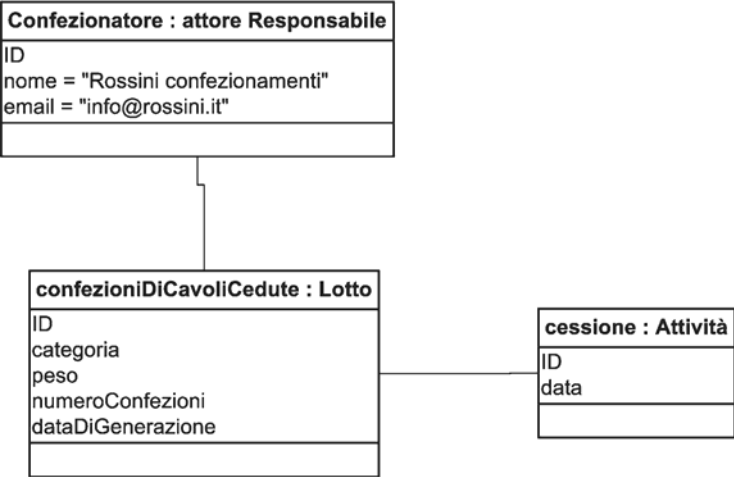


Figura 30 Struttura statica per confezioni di cavolo cedute

Come al solito è necessario registrare il cambio di responsabilità dovuto all’acquisto delle confezioni di cavoli da parte del distributore

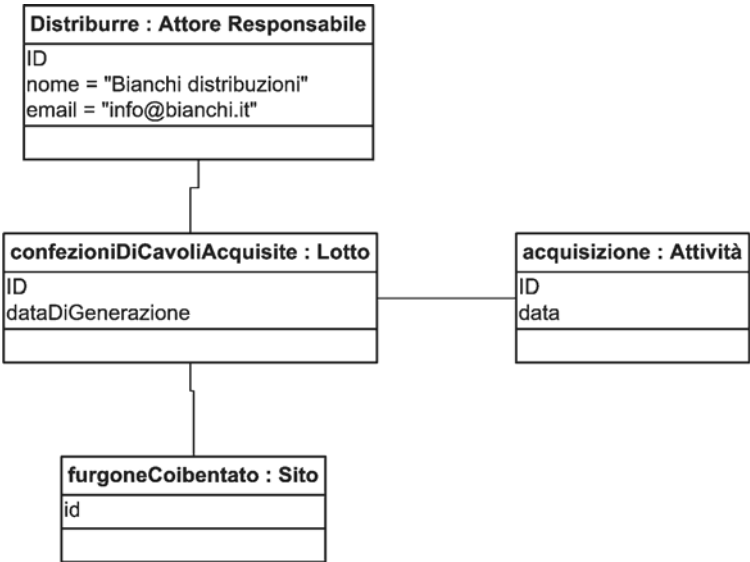


Figura 31 Struttura statica per confezioni di cavolo acquisite

Durante l'attività di *conservazione*, che avviene in appositi magazzini frigoriferi, è chiaro che sarà necessario tenere traccia della *temperatura* di conservazione.

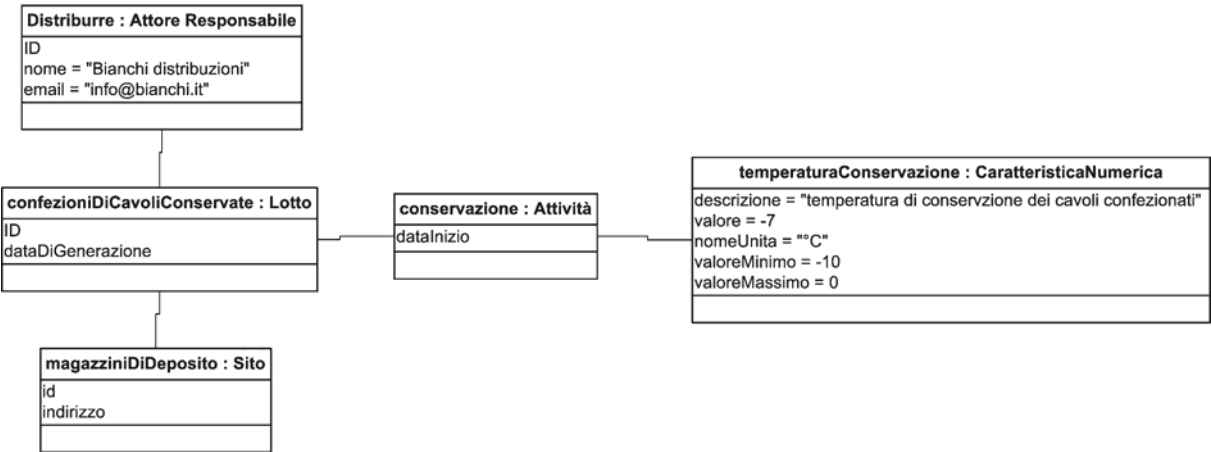


Figura 32 Struttura statica per confezioni di cavoli conservate

Con questo ultimo stato si registra l'avvenuta cessione al consumatore attraverso l'attività di vendita. Questo lotto non presenta caratteristiche rilevanti.

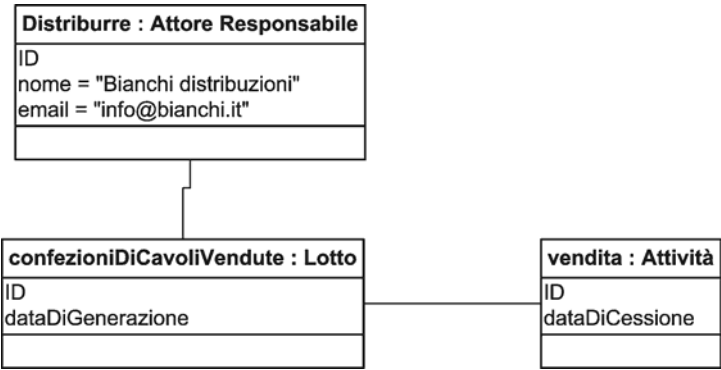


Figura 33 Struttura statica per confezioni di cavoli vendute

2.7 Disciplinare semplificato delle procedure in filiera

In questo capitolo cercheremo di sviluppare altri processi riguardanti la filiera, non solo quelli relativi alla tracciabilità, come ad esempio i processi relativi all'ingresso nella filiera stessa (che supporremo già esistente), o i processi relativi agli ordini di acquisto tra un attore e l'altro. In seguito cercheremo di modellare queste procedure utilizzando lo standard ebXML che offre un valido supporto per transazioni tra aziende, fornendo indiscutibili vantaggi, tra i quali il fatto di essere un protocollo non proprietario, che permette bassi investimenti iniziali, e l'interoperabilità fornita dall'indipendenza da protocolli di trasporto o linguaggi di programmazione permettendo la facile integrazione con eventuali sistemi informatici già presenti. Vi sono due tipi di accordo, il primo tra attore e coordinatore che permette di svolgere le procedure per l'ingresso in filiera, il secondo (dinamico a seconda dei rapporti di mercato) tra attore ed attore.

2.7.1 Attore Responsabile - Coordinatore

- 1) l'Attore responsabile etichetta (fisicamente) i siti, invia i propri dati (codice identificativo, ragione sociale, indirizzo, telefono, e-mail....) e i dati dei propri siti (codice sito, indirizzo, descrizione....) al coordinatore
- 2) il coordinatore invia una ricevuta di ritorno e risponde comunicando la data in cui verrà effettuata la verifica
- 3) il coordinatore esegue l'ispezione per verificare l'esecuzione effettiva del punto 1 e per disporre le infrastrutture tecnologiche da inserire
- 4) il coordinatore comunica l'esito della verifica.
in caso di fallimento della verifica il coordinatore invia le modifiche da effettuare e propone una nuova ispezione.
In caso di successo della verifica il coordinatore invia un documento con le attività di aggiornamento che l'attore dovrà eseguire)
- 5) l'attore invia la ricevuta e risponde quando l'aggiornamento è stato fatto
- 6) il coordinatore comunica la ricezione della comunicazione.

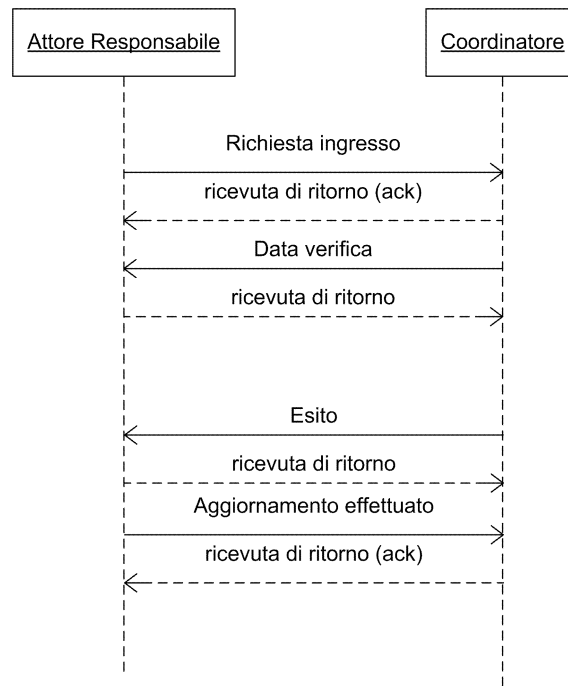


Figura 34 Diagramma di sequenza ingresso in filiera

2.7.2 Coltivatore – Fornitore Piante

- 1) Il Coltivatore sceglie un Fornitore Piante per istaurare un rapporto commerciale
- 2) Il coltivatore invia un ordine al Fornitore contenente un allegato con le istruzioni di consegna
- 3) Il Fornitore invia entro un giorno dal ricevimento dell'ordine, la ricevuta di ritorno dell'ordine stesso, ed entro 2 giorni la risposta all'ordine che specifica se l'ordine può essere soddisfatto completamente, parzialmente, o non possa essere evaso.
- 4) Nel caso del superamento dei 2 giorni dall'inizio della transazione la risposta è considerata negativa e l'ordine rifiutato.
- 5) Il coltivatore comunica il ricevimento della risposta entro 1 giorno. Se questa è positiva, deve essere allegato anche un attestato di avvenuto pagamento.
- 6) Il produttore comunica l'avvenuta ricezione delle informazioni relative al pagamento

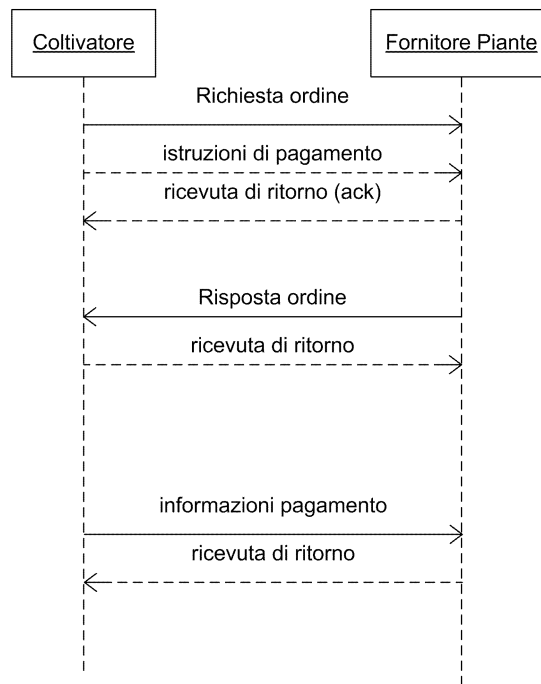


Figura 35 Diagramma di sequenza acquisto piante

2.7.3 Coltivatore - Confezionatore

- 1) Il coltivatore cerca un confezionatore per istaurare un rapporto commerciale .
- 2) Il coltivatore comunica al confezionatore i parametri (prezzo, peso previsto, consistenza, danneggiamento...) del raccolto che si prepara ad effettuare.
- 3) Il confezionatore risponde con una ricevuta di ritorno entro un giorno e con una risposta di accettazione o rifiuto della merce entro due giorni
- 4) Il coltivatore risponde con una ricevuta di ritorno entro un giorno.
- 5) In caso di accettazione del raccolto da parte del confezionatore, il coltivatore comunica, entro cinque giorni dal ricevimento della conferma, la data e l'ora prevista per l'inizio della raccolta e l'ora e il luogo in cui il confezionatore dovrà recarsi per caricare la merce.
- 6) Il confezionatore risponde con una ricevuta di ritorno entro un giorno.
- 7) Il confezionatore una volta accettato l'ordine (ed eventualmente verificato la conformità) deve provvedere al pagamento ed inviare un messaggio al coltivatore con in allegato una ricevuta di avvenuto pagamento entro un mese dall'accettazione.
- 8) Il coltivatore risponde con un ricevuta entro 2 giorni

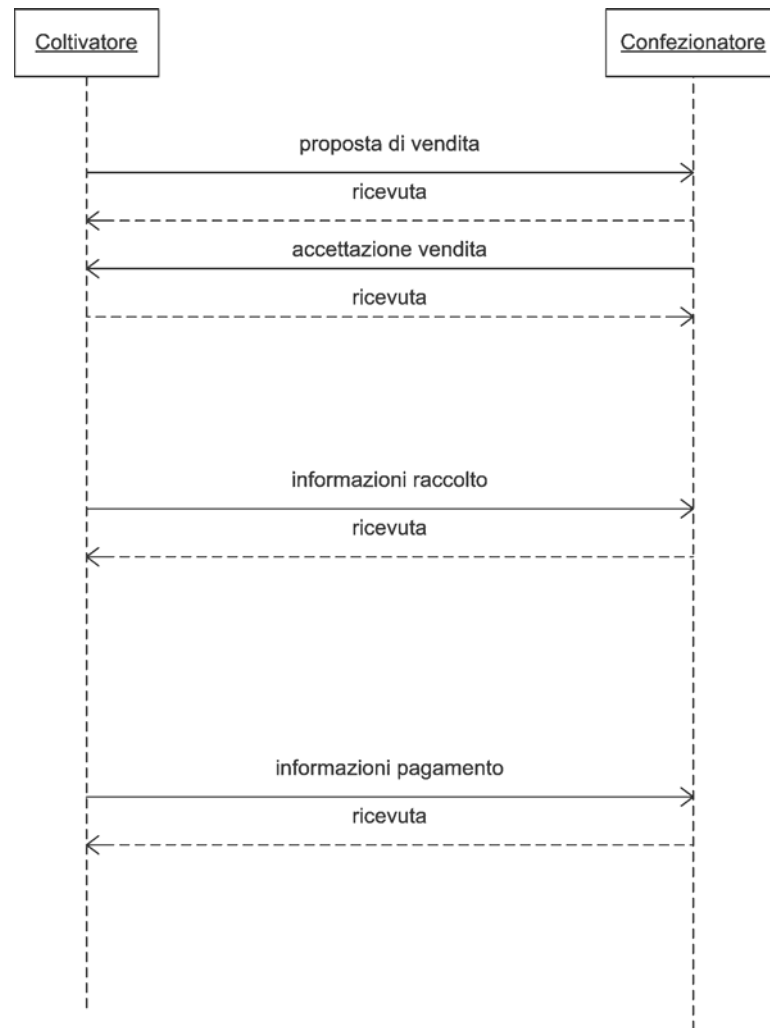


Figura 36 Diagramma di sequenza vendita cavoli

2.7.4 Confezionatore - Distributore

- 1) Il Confezionatore cerca un distributore per istaurare un rapporto commerciale.
- 2) Il Confezionatore fa una proposta di vendita al distributore, comunicandogli le caratteristiche della merce che intende vendere (peso, categoria,...).
- 3) Il distributore ha 1 giorno per confermare (con una ricevuta di ritorno) e per rispondere se intende accettare o rifiutare la merce.
- 4) In caso di risposta affermativa il distributore invia anche le informazioni del luogo in cui desidera ricevere la merce.
- 5) Il confezionatore invia una risposta entro un giorno.
- 6) Entro venti giorni da quando il distributore accetta la proposta, questo deve inviare un messaggio con allegato una certificazione di pagamento.
- 7) Il confezionatore invia una ricevuta della ricezione del pagamento.

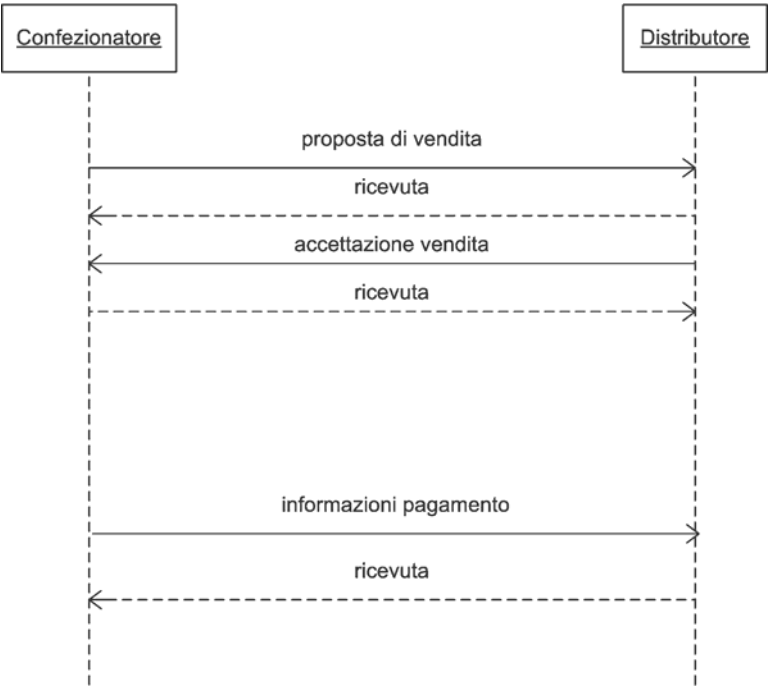


Figura 37 Diagramma di sequenza vendita confezioni di cavoli

Capitolo 3

Lo standard ebXML

In questo capitolo viene presentato lo standard ebXML, paradigma per il commercio elettronico. Lo stesso viene poi usato per implementare alcune delle procedure modellate in precedenza.

3.1 L'architettura ebXML

ebXML è un insieme di specifiche pensate per permettere alle aziende di gestire le proprie Transazioni via Internet in modo facile ed efficiente.

ebXML fornisce alle aziende una semantica comune per creare un'infrastruttura per lo scambio di messaggi aziendali, la gestione di rapporti commerciali e la comunicazione di dati, in maniera automatica, definendo e registrando processi aziendali quali l'evasione degli ordini, la spedizione, ecc...

Questo standard permette una facile integrazione con eventuali sistemi già presenti, ma soprattutto offre il vantaggio di richiedere sforzi economici pressoché nulli e conoscenze informatiche non eccessive, rendendosi adatto dunque anche alle piccole e medie imprese.

Il sistema di messaggistica di ebXML utilizza messaggi XML e li mappa su SOAP, il quale garantisce la compatibilità con diversi protocolli di basso livello quali HTTP, FTP e SMTP.

Queste due tecnologie permettono una forte interoperabilità del sistema, necessaria quando si deve utilizzare la rete ed è possibile avere a che fare con mondi software o hardware molto disparati.

Uno degli obiettivi principali di ebXML, ed anche uno dei suoi punti di forza, è quello di permettere, anzi di preferire, soluzioni software non proprietarie e renderle indipendenti dal linguaggio.

ebXML è ancora in fase di standardizzazione sotto il controllo di due organizzazioni: OASIS che si occupa dello sviluppo delle specifiche tecniche (infrastruttura) e UN/CEFACT che ha il compito di sviluppare, approvare e mantenere il contenuto di business ebXML-compliant.

3.1.1 Elementi principali di ebXML

Nell'architettura ebXML vengono definiti alcuni concetti principali che possiamo riassumere come segue:

- *Business Process*: è la specifica che permette di descrivere le transazioni tra due o più parti, specificando i ruoli, la coreografia delle varie transazioni che compongono il processo di business ed i messaggi che vengono scambiati durante le transazioni.
- *Collaboration Protocol Profile*: specifica il profilo della società, a quali business process è in grado di partecipare e con quali ruoli, dichiara quali protocolli sono supportati e quali messaggi la società è in grado di processare sia in ingresso che in uscita.
- *Collaboration Protocol Agreement*: nasce dalla fusione di due CPP e stabilisce un accordo tra le due aziende sui protocolli da utilizzare, che ovviamente devono essere tra loro compatibili, e sulle modalità per lo scambio dei messaggi.
- *Registry/Repository*: è il raccoglitore pubblico di tutti i documenti ebXML, compresi BP, CPP, CPA ecc..., ma può contenere anche altri documenti relativi alle aziende. Di solito viene implementato con una servlet centrale.
- *Business Document*: sono i documenti scambiati fra le aziende all'interno delle transazioni. Questi documenti sono specifici per determinati contesti di business e possono essere composti da componenti elementari detti *Core Component*.
- *Core Component* sono componenti standardizzati che possono essere utilizzati nella specifica dei documenti ebXML.

3.1.2 Business Process

La piattaforma ebXML permette attraverso una specifica (Business Process Specification Schema) la definizione chiara di processi di business.

Lo schema supporta la specifica di Transazioni di Business (scambi elementari di documenti, formati da una richiesta ed un eventuale risposta) e la loro combinazione all'interno delle Collaborazioni di Business. Questi schemi controllano lo scambio dei Documenti e dei messaggi che i Partner Commerciali si scambiano per portare a termine la transazione di commercio elettronico desiderata.

Lo schema è mantenuto nel *Registry*, che è un contenitore pubblico, e può essere riutilizzato anche da altre aziende.

Un *Business Process* definisce quindi l'ordine con cui vengono inviati i messaggi all'interno di una collaborazione B2B, ma non tiene conto di come vengono creati o elaborati i dati e i messaggi scambiati.

I partecipanti alla transazione assumono dei ruoli stabiliti all'interno dello schema, ruoli a cui corrispondono transazioni prestabilite per lo scambio di documenti. Una collaborazione può essere binaria (tra due parti) o multiparte (tra tre o più parti). Quest'ultima deve essere espressa come un insieme di collaborazioni binarie.

Una transazione consiste in uno o più flussi di comunicazione.

Il *Business Transaction Choreography* è la componente del *Business Process* che si occupa di descrivere l'ordinamento delle transazioni durante le collaborazioni.

Un *Business Process* è costituito dalle seguenti componenti :

- Specifica della transazione
- Specifica del flusso dei documenti
- Specifica della collaborazione binaria e della coreografia delle transazioni all'interno di essa
- Specifica della collaborazione multiparte sviluppandola in collaborazioni binarie

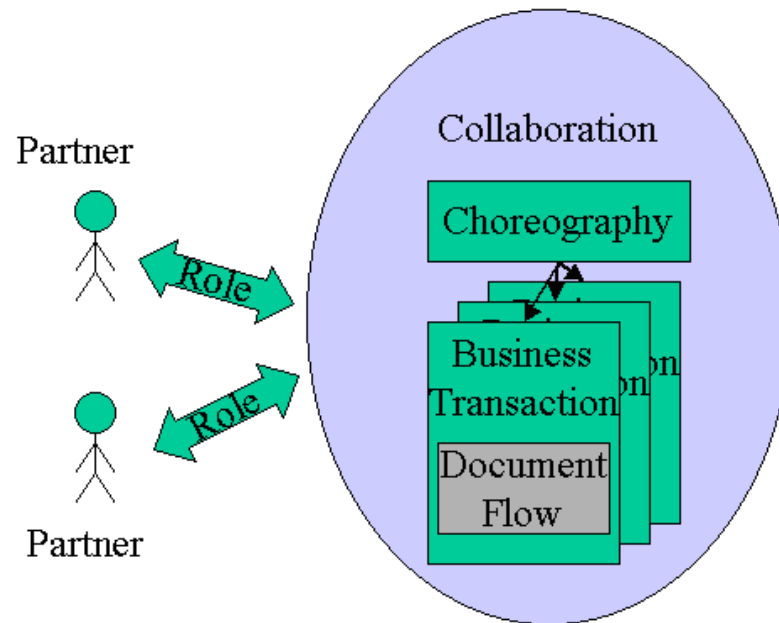


Figura 38 Business collaboration

3.1.3 Collaboration Protocol Profile

Lo scambio di informazioni fra due controparti richiede che ciascuna di esse sia a conoscenza delle capacità di business supportate dall'altra.

Queste riguardano i ruoli supportati all'interno delle collaborazioni ed i dettagli tecnologici circa il modo in cui tale parte invia e riceve i messaggi, come ad esempio i protocolli supportati.

In alcuni casi, è necessario che le due controparti raggiungano un accordo su tali dettagli.

Per facilitare questo scambio di informazioni, ebXML descrive il Collaboration Protocol Profile (CPP) ed il Collaboration Protocol Agreement (CPA).

Il primo descrive tutte le potenzialità ed i modi in cui una parte può impegnarsi in un business elettronico con altre parti. Queste potenzialità sono sia tecnologiche (protocolli supportati e modalità di scambio dei messaggi) sia commerciali (quali processi di business supporta e con quali ruoli). Una azienda può descrivere se stessa in un unico CPP, oppure creare differenti CPP che descrivono le sue operazioni in varie regioni del mondo, o le diverse sezioni della propria organizzazione.

Per facilitare l'instaurarsi di collaborazioni commerciali questi documenti possono essere immagazzinati in un appositi contenitori pubblici, implementati ad esempio con l'ebXML Registry (ebRS).

Questi Repository forniscono funzioni di ricerca ed individuazione, codificate nelle specifiche del repository, attraverso le quali una parte può trovare adeguati partner commerciali.

Di seguito viene riportato un esempio di CPP:

```
<tp:CollaborationProtocolProfile
  xmlns:tp="http://www.ebxml.org/namespaces/tradePartner"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.ebxml.org/namespaces/tradePartner
http://ebxml.org/project_teams/trade_partner/cpp-cpa-v1_0.xsd"
  tp:version="1.1">
  <tp:PartyInfo>
    <tp:PartyId>123456789</tp:PartyId>
    <tp:PartyRef tp:href="http://www.azienda-A.it/info.html"/>
    <tp:CollaborationRole tp:id="N00">
      <tp:ProcessSpecification tp:name="buySell" ...
        xlink:href="www.ebxml.org/buy.xml">
        ...
      </tp:ProcessSpecification>
      <tp:Role tp:name="acquirente" xlink:href="..."/>
      <tp:CertificateRef tp:certId="N03"/>
      <tp:ServiceBinding tp:channelId="N04" tp:packageId="N0402">
        ...
      </tp:ServiceBinding>
    </tp:CollaborationRole>
    <tp:Certificate tp:certId="N03">...</tp:Certificate>
    <tp:DeliveryChannel tp:channelId="N04" tp:transportId="N05"
      tp:docExchangeId="N06">
      ...
    </tp:DeliveryChannel>
    <tp:Transport tp:transportId="N05">
      ...
    </tp:Transport>
    <tp:DocExchange tp:docExchangeId="N06">
      <tp:ebXMLBinding tp:version="0.98b">
        ...
      </tp:ebXMLBinding>
    </tp:DocExchange>
  </tp:PartyInfo>
  <tp:SimplePart tp:id="  ">
    ...
  </tp:SimplePart>
  <tp:Packaging tp:id="N0402">
    ...
  </tp:Packaging>
  <tp:Comment tp:xml_lang="en-us">Accordo di vendita</tp:Comment>
</tp:CollaborationProtocolProfile>
```

Esso contiene uno o più campi *PartyInfo* che descrivono l'azienda (o parti di essa), quali processi di business (*ProcessSpecification*) implementa ed i ruoli supportati all'interno di questi ultimi (*Role*). Vengono specificati i protocolli di trasporto supportati (*Transport*) e le varie codifiche possibili per lo scambio di messaggi (*DocExchange*). Si specifica infine come viene "impacchettata" l'informazione da trasmettere attraverso i campi *SimplePart* ed i campi *Packaging*.

3.1.4 Collaboration Protocol Agreement

Il *Collaboration Protocol Agreement* (CPA) definisce l'accordo che le due controparti devono raggiungere prima di potersi impegnare in transazioni di business.

Un CPA è solitamente creato mediante elaborazioni e negoziazioni successive derivanti dall'incrocio di due CPP. Ad esempio: un CPA include solamente quegli elementi che sono comuni o compatibili fra i CPP delle due parti.

Il CPA è indipendente dai particolari processi interni attuati da ciascuna parte, che vengono interfacciati verso l'esterno tramite le *Business Collaboration* descritte dal CPA e dal documento di Process-Specification (descritto nel BPSS). In tal modo non vengono esposti alla controparte dettagli dei processi interni.

Il CPA fornisce una specifica ad alto livello che può essere facilmente comprensibile da persone ma allo stesso tempo rigorosa e precisa da poter essere usata in maniera automatica.

I CPA sono documenti elettronici in formato XML e codifica testuale, e possono essere processati direttamente dai sistemi delle controparti per preparare ed eseguire gli scambi di informazioni voluti.

Eventuali termini e condizioni legali di un accordo commerciale sono al di là degli scopi di queste specifiche e non sono inclusi né nel CPP né nel CPA, anche se esso fornisce già una base, prevedendo caratteristiche quali la non-ripudiabilità e la firma elettronica dei messaggi.

Segue un esempio che presenta la struttura di un CPA.

```
<CollaborationProtocolAgreement
  xmlns="http://www.ebxml.org/namespaces/tradePartner"
  cpaId="YoursAndMyCPA"
  version="1.2">
  <Status value="proposed"/>
  <Start>1988-04-07T18:39:09</Start>
  <End>1990-04-07T18:40:00</End>
  <ConversationConstraints invocationLimit="100"
    concurrentConversations = "4"/>
  <PartyInfo>
    ...
  </PartyInfo>
  <PartyInfo>
    ...
  </PartyInfo>
  <Packaging id="N20">
    ...
  </Packaging>
  <ds:Signature>any combination of text and elements</ds:Signature>
  <Comment xml:lang="en-gb">any text</Comment>
</CollaborationProtocolAgreement>
```

Dall'esempio viene evidenziata la presenza di due campi *PartyInfo* che rappresentano le due controparti dell'accordo. Vengono quindi indicati i protocolli di trasporto e di sicurezza scelti

e i ruoli assunti dalle due aziende. Ovviamente i protocolli delle due controparti devono essere compatibili.

3.1.5 Message Service

Negli ultimi anni abbiamo assistito alla proliferazione di protocolli per gli scambi commerciali attraverso la rete internet.

Questi protocolli si basano generalmente su EDI (Electronic Data Interchange) che, pur essendosi rivelati efficaci, presentano svantaggi di essere tecnologie proprietarie, e quindi costose, e spesso legate a protocolli di comunicazione specifici.

I principali obiettivi di ebXML a livello di servizio di messaggistica:

- Facilitare l'interscambio di messaggi di business
- Essere neutrale rispetto al protocollo usato
- Essere neutrale rispetto al contenuto del messaggio
- Essere sicuro
- Poter garantire l'affidabilità di recapito dei messaggi.

Basandosi su questi obiettivi, ebXML ha prodotto la *ebXML Message Service Specification* che definisce una specifica per il trasporto dei messaggi in maniera affidabile e sicura.

I messaggi sono caratterizzati dai seguenti punti:

- Sono scritti in formato XML, più precisamente usando il protocollo SOAP with Attachment
- Non sono legati ad alcun protocollo di trasporto.
- I Documenti scambiati tramite la specifica ebXML, identificati dal corpo (payload) dei messaggi ebXML, non sono necessariamente espressi in XML. Questi possono assumere qualsiasi formato digitale (XML, immagini, file di testo, tabelle di database, file binari, ecc.).
- Vengono utilizzate tutte le attuali tecnologie per la sicurezza quali Firme Digitali e Cifratura a Chiave Pubblica.
- Viene implementato un meccanismo basato su segnali di acknowledgement e Timeouts per garantire la ricezione dei messaggi entro i limiti temporali prestabiliti.

L'ebXML Message Service può essere concettualmente suddiviso in tre parti:

- Una Interfaccia di Servizio (*Message Service Interface*).
- Funzioni fornite dal Gestore del Servizio Messaggi (Message Service Handler, MSH).
- Un sistema di mapping verso il protocollo scelto per di trasporto dei messaggi.

La *Message Service Interface* dispone delle seguenti funzioni :

- *Send*: spedisce un messaggio ebXML, i valori per i parametri sono ricavati dal *Message Header*.
- *Receive*: ricezione dei messaggi
- *Notify*: fornisce notifica di eventi attesi e inattesi.
- *Inquire*: fornisce un metodo per conoscere lo stato di un particolare scambio di messaggi.

3.1.6 Registry

Il Registry fornisce una locazione stabile in cui rendere persistenti informazioni depositate dalle aziende. Tali informazioni vengono usate per facilitare partnership e transazioni Business to Business (B2B) basate su ebXML. Il materiale sottoposto può consistere di schemi e documenti XML, tra i quali descrizioni di processi, *ebXML Core Component*, descrizioni di contesto, ma anche di modelli UML, documenti elettronici, informazioni varie circa le parti commerciali e persino componenti software.

Secondo le specifiche ebXML l'architettura del *Registry* consiste di due elementi: il *Registry Service* ed i *Registry Client*. Il primo è preposto alla gestione di un repository, compreso il management delle politiche di pubblicazione e di accesso; il Registry Client è invece un'applicazione usata per accedere al *Registry*.

3.1.7 Core Components

Così come è emerso il bisogno di una procedura standard per definire Processi di Business, allo stesso modo risulta molto utile definire e creare una semantica comune, al fine di evitare incomprensioni dovute all'uso di una terminologia comune ma a cui vengano associati significati diversi.

Il requisito principale risulta ancora una volta quello di garantire l'interoperabilità fra diverse organizzazioni.

Per questo ebXML propone una metodologia per sviluppare un insieme comune di componenti semantici elementari, che rappresentano i tipi generali di dati e di vocabolari di

business correntemente in uso e che poi possano essere riusati e combinati per descrivere transazioni di business.

Usare i *Core Component* garantisce che due diversi partner commerciali che usano differenti sintassi, utilizzino la semantica di business allo stesso modo; inoltre permette un riuso efficiente delle semantiche di business all'interno del medesimo settore di produzione.

3.1.8 Il funzionamento di ebXML

In questo paragrafo si riassume con un esempio a livello di business il funzionamento dell'intero paradigma.

Ogni partecipante definisce un profilo che descrive i messaggi supportati ed il loro formato, i protocolli di trasporto usati e le eventuali restrizioni. Questi profili vengono depositati nel *Registry* consultabile da tutte le aziende.

I profili di due o più aziende partner possono essere combinati per generare un CPA che definisce i messaggi e i protocolli concordati tra le parti, che rappresentano un sottoinsieme di quelli presenti nel documento di specifica del profilo. Inoltre i vari partner si accordano su un *Business Process* predefinito che diventa parte dell'accordo. Quest'ultimo definisce la sequenza con cui questi messaggi vengono inviati e ricevuti e il tipo di documenti inviati come payload.

L'architettura di ebXML può essere dedotta dall'analisi di questo modello.

Analizziamo i vari passi:

- 1) La società A viene a conoscenza dell'esistenza di un Registro ebXML accessibile via Internet.
- 2) La società A interroga il Registry ebXML decide per trovare modelli adatti al proprio business e sviluppa la propria applicazione secondo quanto trovato. Lo sviluppo di software non è essenziale per la partecipazione ad ebXML, in quanto è possibile interfacciare il software eventualmente già esistente, o, eventualmente, sfruttare programmi esistenti adatti allo scopo (vedi capitoli seguenti)
- 3) La società A crea il proprio profilo commerciale (*CPP*) al *Registry*. Quest'ultimo descrive tutti gli scenari di business che essa supporta e le capacità e vincoli nel fare ciò.
- 4) La società B scopre gli scenari di business supportati dalla Società A nel *Registry*.
- 5) La società B invia una richiesta alla società A segnalando l'intenzione di impegnarsi in una transazione commerciale appoggiandosi sull'architettura ebXML. Prima di potersi impegnare in uno scenario di interscambio, le società devono trovare un accordo sulle modalità di effettuare tale interscambio. Tale accordo (cpa) delinea i processi di business ed altre specifiche approvate da ambo le parti. Questo accordo spesso deriva

dall'intersezione delle modalità e delle specifiche tecniche di ciascuna azienda nel compiere la transazione desiderata

- 6) Le società A e B possono ora lanciarsi nell'eBusiness usando ebXML.

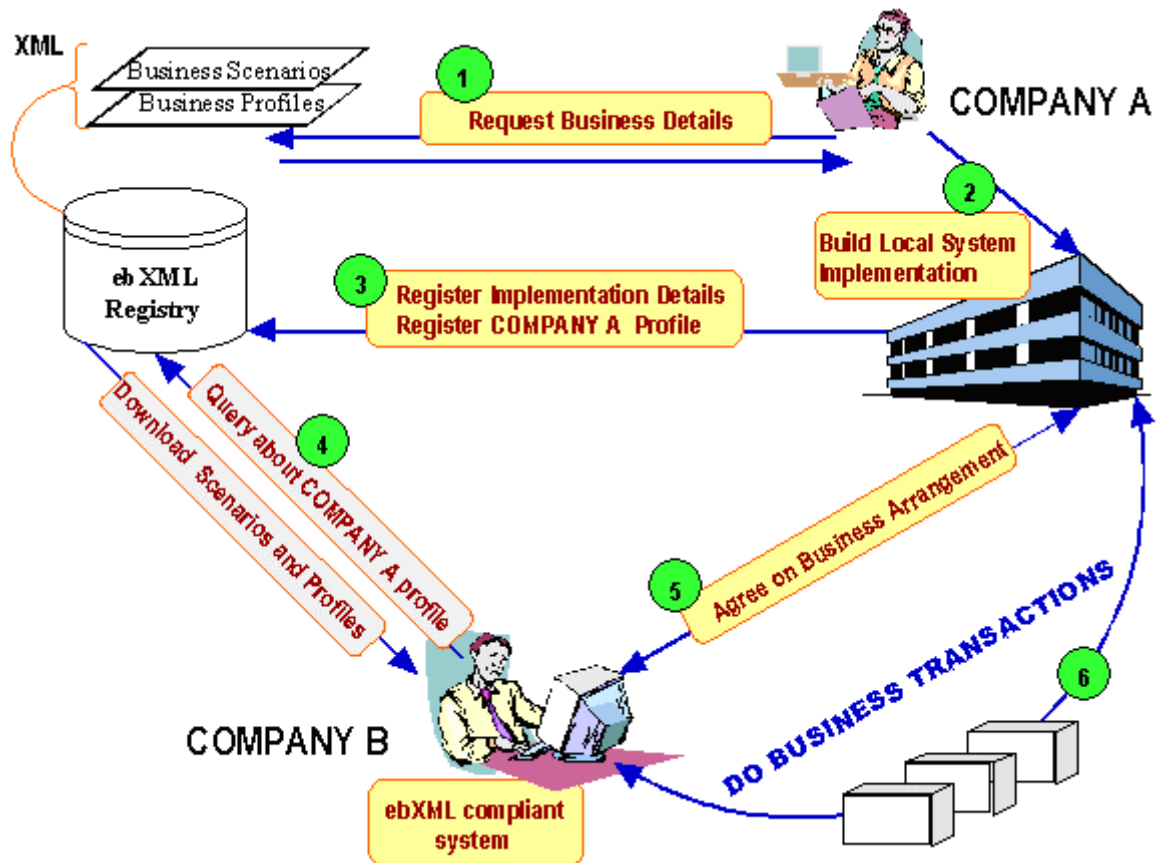


Figura 39 Tipico scenario di utilizzo di ebXML

3.2 Modelli ebXML

Dopo aver visto un'introduzione sugli schemi ebXML necessari per realizzare delle transazioni tra partner commerciali, proviamo a realizzare alcuni di questi schemi per applicarli all'intera filiera da noi analizzata, senza limitarci ai soli processi riguardanti la tracciabilità.

Come esempi prenderemo in considerazione le procedure analizzate per l'ingresso in filiera di una nuova azienda e l'ordine di acquisto piante da parte di un coltivatore nei confronti di un fornitore di piante.

Per queste transazioni realizzeremo dei Business Process ed in seguito dei CPA, cercando di spiegare meglio i vari campi che incontreremo nella loro compilazione.

3.2.1 Analisi Specifica dei campi dei Business Process

Analizziamo ora in dettaglio i vari campi delle due specifiche di business, presenti in forma completa nelle appendici F e G.

Per prima cosa si fornisce un nome, un identificatore ed una versione per la specifica, sotto forma di attributi dell'elemento principale rappresentato da ***ProcessSpecification***. L'ID verrà poi usato per riferirsi alla specifica all'interno dei documenti (ad esempio dal CPA).

Attraverso l'elemento ***Documentation***, se presente, si possono fornire commenti e spiegazioni riferiti al BP.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification >
<ProcessSpecification
  name="ingresso filiera"
  version="1.2"
  uuid="urn:TraceService">
  <Documentation>questo business process regola l'ingresso in filiera
da parte di un attore responsabile,
  lo scambio di documenti avviene tra l'attore responsabile stesso e il
coordinatore della filiera</Documentation>
```

Si passa poi alla dichiarazione dei ***BusinessDocument*** che non sono altro che i documenti che vengono scambiati durante le transazioni, sia sotto forma di documenti principali, sia sotto forma di attachments. Anche per questi si forniscono gli attributi nome e ID ed è possibile specificare il sito in cui si trova lo schema per questo documento attraverso l'attributo ***specificationLocation***.

```
<BusinessDocument
  name="info richiesta"
  nameID="ir1"/>

<BusinessDocument
  name="esito verifica"
  nameID="ev1" />

<BusinessDocument
  name="data verifica"
  nameID="dv1" />

<BusinessDocument
  name="modifiche effettuate"
  nameID="me1" />
```

Una volta dichiarati tutti i documenti si passa alla descrizione delle **Binary Collaboration** che in pratica organizzano e coordinano le varie **Business Transaction**;

Queste ultime rappresentano le unità atomiche della specifica, oltre a definire un protocollo di interazione tra due ruoli autorizzati.

L'elemento **Package** ha lo scopo di raggruppare le varie attività e collaborazioni in maniera gerarchica.

Come attributi della binary collaboration è possibile specificare delle precondizioni e delle postcondizioni, degli eventi di trigger per l'inizio e la fine della collaborazione stessa, il vincolo di tempo in cui la collaborazione deve concludersi, oltre ovviamente al nome e all'id della collaborazione stessa.

Inoltre devono essere presenti obbligatoriamente elementi figli che specificano il ruolo di iniziatore e di risponditore della collaborazione.

All'interno di una **BinaryCollaboration** c'è la dichiarazione delle **BusinessTransaction** che verranno utilizzate ed il ruolo di richiedente e risponditore della transazione, che non devono necessariamente corrispondere a quello di iniziatore e risponditore della collaborazione; tutto ciò viene dichiarato tramite gli elementi **BusinessTransactionActivity** ed i suoi attributi **businessTransaction**, **fromAuthorizedRole** e **toAuthorizedRole**.

Sempre all'interno di una Binary Collaboration, attraverso elementi figli quali **Start**, **Transition**, **Success**, **Failure**, ecc si specifica la coreografia tra le varie transazioni richiamate all'interno della collaborazione.

```
<Package
  name="ingresso filiera">
  <BinaryCollaboration
    name="richiesta ingresso"
    timeToPerform="P3D">
    <InitiatingRole
      name="richiedente" />

    <RespondingRole
```

```
name="coordinatore" />

<BusinessTransactionActivity
name="richiesta ingresso"
businessTransaction="richiesta ingresso"
fromAuthorizedRole="richiedente"
toAuthorizedRole="coordinatore" />

<BusinessTransactionActivity
name="invio esito"
businessTransaction="invio esito"
fromAuthorizedRole="coordinatore"
toAuthorizedRole="richiedente" />

<Start toBusinessState="richiesta ingresso"/>
<Transition
formBusinessState="richiesta ingresso"
toBusinessState="invio esito"/>

<Success formBusinessState="invio esito"
conditionGuard="Success"/>

</BinaryCollaboration>
```

Infine sono presenti le definizioni delle **BusinessTransaction**, che sono, come già detto le unità atomiche di business.

Esse possono o avere successo o fallire completamente, e nel caso di fallimento la transazione subisce rollback ed è come se non fosse mai iniziata.

Casi di fallimento di una transazione possono essere ad esempio dei documenti scambiati non corretti o non validi, lo scadere dei timeouts....

Ogni transazione contiene, oltre all'attributo nome che serve da riferimento per la transazione all'interno della specifica, contiene due elementi figli che sono **RequestingBusinessActivity** e **RespondingBusinessActivity**.

Ciascuno di questi due elementi può contenere una serie di attributi per specificare le caratteristiche dello scambio di documenti, tra le quali se è richiesto ed entro quanto tempo è richiesto un acknowledgment, se la transazione deve essere non ripudiabile, se i documenti prima di essere confermati devono essere valicati, se è richiesta un autorizzazione...

Ovviamente i due elementi precedenti possono contenere anche il **DocumentEnvelope** che rappresenta la "busta elettronica" che contiene il **BusinessDocument** e gli eventuali **Attachments** correlati all'attività di richiesta o di risposta.

È possibile definire tramite alcuni attributi le caratteristiche del **DocumentEnvelope**, e quindi del **businessDocument**, e degli allegati, anche se per questi è possibile singolarmente una ridefinizione degli attributi che sovrascrive quella del DocumentEnvelope (ovviamente solo per l'attributo).


```
<BusinessTransaction
  name="ordine acquisto">
  <RequestingBusinessActivity
    name=""
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="P1D"
    timeToAcknowledgeAcceptance="P2D">
    <DocumentEnvelope
      isPositiveResponse="true"
      businessDocument="ordine">
      <Attachment
        name="note di consegna"
        businessDocument="info recapito" />
      </DocumentEnvelope>
    </RequestingBusinessActivity>

    <RespondingBusinessActivity
      name=""
      isNonRepudiationRequired="true"
      timeToAcknowledgeReceipt="P1D">
      <DocumentEnvelope
        isPositiveResponse="true"
        businessDocument="esito ordine" />
      </RespondingBusinessActivity>
    </BusinessTransaction>
```

3.2.2 Analisi specifica dei campi dei CPA

In questo capitolo analizzeremo più in dettaglio i campi contenuti nei CPA.

Innanzitutto partiamo dall'analizzare la struttura generale di un CPA che è così formata:

```
<tp:CollaborationProtocolAgreement
  xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-
2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  tp:cpaid="uri:companyAandCompanyBCPA"
  tp:version="2_0b">
  <tp:Status tp:value="signed"/>
  <tp:Start>2005-05-20T07:21:00Z<tp:Start>
  <tp:End>2007-05-20T07:21:00Z<tp:End>
  <tp:ConversationConstraints tp:invocationLimit="100"
tp:concurrentConversation="10">
  <tp:PartyInfo> <!-- one or more -->
...
  </tp:PartyInfo>
  <tp:SimplePart id="..."> <!-- one or more -->
...
  </tp:SimplePart>
  <tp:Packaging id="..."> <!-- one or more -->
...
  </tp:Packaging>
```

```
<tp:Signature> <!-- zero or one -->
...
</tp:Signature>
<tp:Comment>text</tp:Comment> <!-- zero or more -->
</tp:CollaborationAgreement>
```

Si nota subito l'elemento principale che è rappresentato da ***CollaborationProtocolAgreement*** il quale racchiude tutto il documento e presenta, oltre alla dichiarazione dei vari namespaces, degli attributi per specificare la versione utilizzata e l'ID del documento. Sono presenti inoltre degli elementi figli che stabiliscono lo ***status*** del documento (proposed, signed...) la validità temporale del documento, le chiamate contemporanee al documento.

Un CPA è composto inoltre da almeno due elementi ***PartyInfo***, che rappresentano le parti coinvolte nel CPA ed i loro ruoli all'interno della collaborazione, da uno o più campi ***SimplePart***, che rappresentano le parti base dei messaggi, e da uno o più elementi ***Package***, che rappresentano la composizione vera e propria dei messaggi usando una composizione di parti base.

Possono inoltre essere presenti dei campi ***Comment***, che racchiudono eventuali note o commenti, ed un campo ***Signature*** per l'eventuale firma elettronica del CPA stesso.

Ora vediamo più in dettaglio i vari elementi figli.

L'elemento ***PartyInfo*** è composto da alcuni attributi, quali il nome ed il canale e l'impacchettamento del messaggio di default.

```
<!-- Party info for CompanyB -->
<tp:PartyInfo
  tp:partyName="Vivai Landi"
  tp:defaultMshChannelId="asyncChannelB1"
  tp:defaultMshPackageId="CompanyB_MshSignalPackage">
  <tp:PartyId>VL00001</tp:PartyId>

  <tp:PartyRef
    xlink:type="simple"
    xlink:href="http://www.landivivai.it/info.html" />

  <tp:CollaborationRole>
    <tp:ProcessSpecification
      tp:version="1.2"
      tp:name="acquisto piante"
      xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s2.xml"
      tp:uuid="urn:TraceService" />

    <tp:Role
      tp:name="venditore"
      xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s2.xml#venditore" />
```

Gli elementi figli forniscono l'ID dell'azienda (campo **PartyId**), un link a risorse esterne riguardanti la parte (ad esempio il sito aziendale), il ruolo all'interno della collaborazione, i certificati sia dell'azienda sia di autorità fidate esterne, informazioni sui protocolli di trasporto supportati ed infine informazioni riguardanti lo scambio di messaggi.

Più in particolare l'elemento **CollaborationRole** specifica a quale business process ci stiamo riferendo e dove si trova fisicamente questo schema (**ProcessSpecification**), quale ruolo del business process viene svolto dall'azienda (**Role**) ma in particolare fornisce il legame (ServiceBinding, CanSend, CanReceive) tra i messaggi da inviare (**packageId**) ed i canali di comunicazione (**ChannelId**) disponibili propri e della controparte.

```
<tp:ServiceBinding>
  <tp:Service>bpid:icann:iet.unipi.it:3A4$2.0</tp:Service>

  <tp:CanSend>
    <tp:ThisPartyActionBinding
      tp:id="companyA_ABID1"
      tp:action="info richiesta"
      tp:packageId="CompanyA_RequestPackage">
      <tp:BusinessTransactionCharacteristics
        tp:isNonRepudiationRequired="true"
        tp:isConfidential="transient"
        tp:isAuthenticated="persistent"
        tp:isTamperProof="none"
        tp:isAuthorizationRequired="true"
        tp:timeToAcknowledgeReceipt="PT2H" />

      <tp:ActionContext
        tp:binaryCollaboration="richiesta ingresso"
        tp:businessTransactionActivity="richiesta ingresso"
        tp:requestOrResponseAction="a" />

      <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

    <tp:OtherPartyActionBinding>companyB_ABID4</tp:OtherPartyActionBinding>
```

Ovviamente queste informazioni devono essere le stesse, o almeno compatibili, negli altri attori della collaborazione.

Inoltre è possibile specificare le caratteristiche della transazione di business, quali la non ripudiabilità, l'autenticazione, la non modificabilità, il crittaggio, la persistenza dei dati e le scadenze entro le quali ci aspettiamo di concludere la transazione, il tutto attraverso l'elemento **BusinessTransactionCharacteristics**.

Da notare che, poiché un'azienda può svolgere dei ruoli in più di una business collaboration o può avere ruoli diversi in una stessa collaborazione, ciascun elemento PartyInfo potrebbe contenere più di un campo CollaborationRole .

Un altro elemento figlio di PartyInfo interessante è rappresentato da ***DeliveryChannel*** che fornisce tutte le possibili combinazioni supportate dall'attore tra protocolli di trasporto e modi di scambio dei documenti, oltre a dichiarare alcune caratteristiche del canale, come se è richiesto un acknowledgment di conferma, se ci si aspetta una risposta sincrona, se si devono eliminare i duplicati....

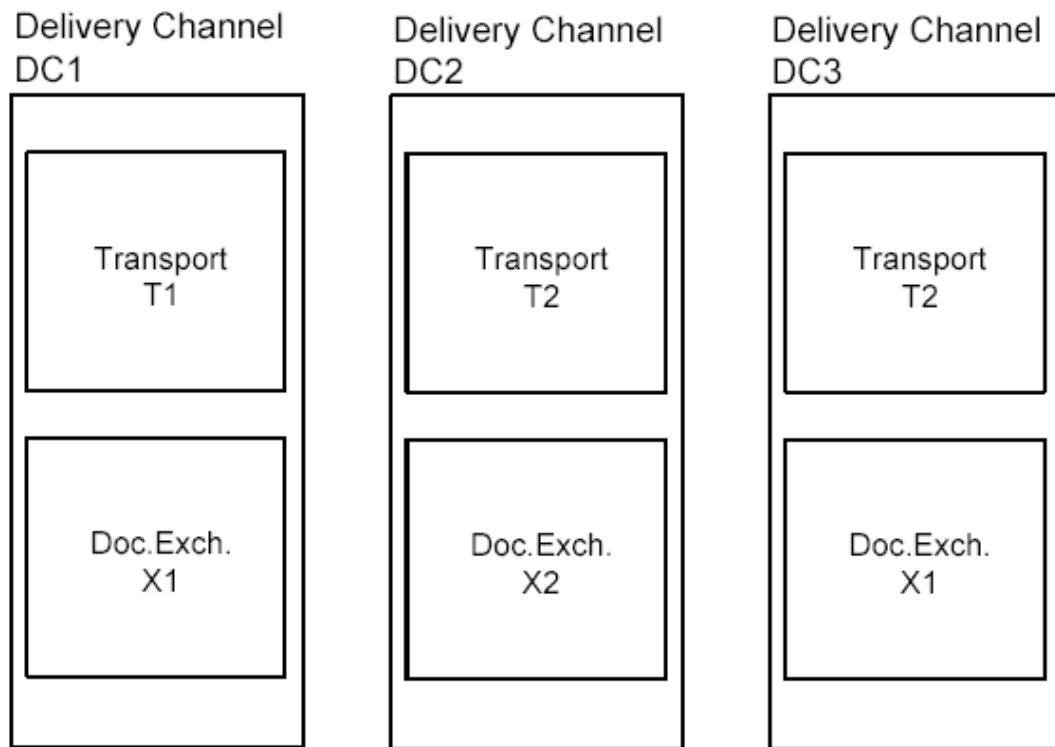


Figura 40 Esempio di composizione di delivery channel

```
<tp:DeliveryChannel
  tp:channelId="asyncChannelA1"
  tp:transportId="transportA1"
  tp:docExchangeId="docExchangeA1">
  <tp:MessagingCharacteristics
    tp:syncReplyMode="none"
    tp:ackRequested="always"
    tp:duplicateElimination="always" />
  </tp:DeliveryChannel>
```

I vari elementi ***Transport*** forniscono la descrizione dei protocolli di trasporto utilizzabili sia nella parte relativa alla ricezione sia nella parte relativa al sending, compresi i protocolli di sicurezza utilizzati, il metodo di accesso, gli endpoints, ecc.....

```
<tp:Transport
  tp:transportId="transportA1">
  <tp:TransportSender>
    <tp:TransportProtocol
      tp:version="1.1">HTTP</tp:TransportProtocol>

    <tp:AccessAuthentication>basic</tp:AccessAuthentication>
```

```
<tp:TransportClientSecurity>
  <tp:TransportSecurityProtocol
    tp:version="3.0">SSL</tp:TransportSecurityProtocol>

  <tp:ClientCertificateRef
    tp:certId="CompanyA_ClientCert" />

  <tp:ServerSecurityDetailsRef
    tp:securityId="CompanyA_TransportSecurity" />
</tp:TransportClientSecurity>
</tp:TransportSender>

<tp:TransportReceiver>
  <tp:TransportProtocol
    tp:version="1.1">HTTP</tp:TransportProtocol>

  <tp:AccessAuthentication>basic</tp:AccessAuthentication>

  <tp:Endpoint
    tp:uri="https://www.landivivai.it/servlets/ebxmlhandler/async"
    tp:type="allPurpose" />

  <tp:TransportServerSecurity>
    <tp:TransportSecurityProtocol
      tp:version="3.0">SSL</tp:TransportSecurityProtocol>

    <tp:ServerCertificateRef
      tp:certId="CompanyA_ServerCert" />

    <tp:ClientSecurityDetailsRef
      tp:securityId="CompanyA_TransportSecurity" />
  </tp:TransportServerSecurity>
</tp:TransportReceiver>
</tp:Transport>
```

I vari elementi **DocExchange** invece trattano le proprietà che devono avere i vari scambi di messaggi, quali possedere o meno caratteristiche di messaggistica affidabile (numero di tentativi e intervallo di tempo tra un tentativo e l'altro), garantire o meno l'ordine dei messaggi, la durata della persistenza all'interno del message service, la non ripudiabilità (protocolli utilizzati per l'hashing e per la firma digitale), l'Encryption algorithm utilizzato.....

```
<tp:DocExchange
  tp:docExchangeId="docExchangeA1">
  <tp:ebXMLSenderBinding
    tp:version="2.0">
    <tp:ReliableMessaging>
      <tp:Retries>3</tp:Retries>

      <tp:RetryInterval>PT2H</tp:RetryInterval>

    </tp:ReliableMessaging>
  </tp:ebXMLSenderBinding>
  <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
  <tp:PersistDuration>P1D</tp:PersistDuration>
  <tp:SenderNonRepudiation>
```

```
<tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudiationProtocol>
```

```
<tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>
```

```
<tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-sha1</tp:SignatureAlgorithm>
```

```
    <tp:SigningCertificateRef
      tp:certId="CompanyA_SigningCert" />
  </tp:SenderNonRepudiation>
```

```
  <tp:SenderDigitalEnvelope>
    <tp:DigitalEnvelopeProtocol
      tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>
```

```
    <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>
```

```
    <tp:EncryptionSecurityDetailsRef
      tp:securityId="CompanyA_MessageSecurity" />
  </tp:SenderDigitalEnvelope>
```

```
</tp:ebXMLSenderBinding>
```

```
<tp:ebXMLReceiverBinding
  tp:version="2.0">
  <tp:ReliableMessaging>
    <tp:Retries>3</tp:Retries>
```

```
    <tp:RetryInterval>PT2H</tp:RetryInterval>
```

```
<tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
</tp:ReliableMessaging>
```

```
<tp:PersistDuration>P1D</tp:PersistDuration>
```

```
<tp:ReceiverNonRepudiation>
```

```
<tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</
tp:NonRepudiationProtocol>
```

```
ion>
  <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>
```

```
  <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-sha1</tp:SignatureAlgorithm>
```

```
    <tp:SigningSecurityDetailsRef
      tp:securityId="CompanyA_MessageSecurity" />
  </tp:ReceiverNonRepudiation>
```

```
<tp:ReceiverDigitalEnvelope>
  <tp:DigitalEnvelopeProtocol
    tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>
    <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>
    <tp:EncryptionCertificateRef
      tp:certId="CompanyA_EncryptionCert" />
```

```
  </tp:ReceiverDigitalEnvelope>
```

```
</tp:ebXMLReceiverBinding>
```

```
</tp:DocExchange>
```

Gli elementi ***SimplePart*** rappresentano i “mattoni” base con cui costruire i vari messaggi, identificate attraverso il valore dell’attributo *mimetype*.

3.3 Come utilizzare i documenti prodotti

Come è facilmente intuibile da ciò che è stato introdotto precedentemente, la specifica ebXML non implementa e soprattutto non vincola in alcun modo l’effettiva transazione tra le due parti, ma rappresenta una specie di contratto sulle modalità, sui tempi e sui mezzi tramite cui la transazione può avvenire.

Infatti il BP si limita a fissare i ruoli di un processo di business e stabilire le tempistiche dello scambio di documenti, oltre a definire le tipologie di questi ultimi.

Invece un CPA stabilisce i legami tra aziende e processi implementati, i ruoli effettivamente ricoperti all’interno di questi ultimi, definisce i mezzi tecnologici e le modalità con cui vengono utilizzati per lo svolgimento della transazione tra le due parti.

Questi due modelli possono venir presi come base per implementare una applicazione ad hoc per la transazione, soluzione efficace ma molto costosa ed impegnativa, che non si adatta alle piccole imprese.

Una soluzione alternativa, proponibile per medie imprese che possiedono già dei sistemi informativi, è quella di realizzare delle interfacce tra i software già esistenti ed i sistemi di messaggistica di ebXML.

Ciò comporta, oltre al minor costo di sviluppo, anche un minor tempo di apprendimento da parte degli operatori che si trovano a dover usare una nuova funzionalità anziché un nuovo intero sistema.

Per le piccole imprese, che sono poi quelle che maggiormente caratterizzano la filiera alimentare nel primo stadio della catena produttiva, è possibile affidarsi a delle soluzioni open-source che richiedono sforzi tecnologici ma soprattutto economici quasi nulli. Tutti gli standard di rilievo, come ebXML, forniscono tale supporto software gratuito mediante organizzazioni non profit.

Uno di questi software che andremo ad analizzare nel dettaglio nei prossimi capitoli è rappresentato da ebMail.

3.3.1 ebMail

ebMail è il frutto di un progetto seguito e sviluppato dal CECID e dall'università di Hong Kong e realizza un sistema per il commercio elettronico tramite lo standard ebXML, basato su di un interfaccia grafica tipica di un programma di posta elettronica.

Anche l'invio dei messaggi avviene tramite il protocollo di posta elettronica e cioè SMTP, protocollo supportato dallo standard ebXML. In figura è visibile l'interfaccia grafica di ebMail.

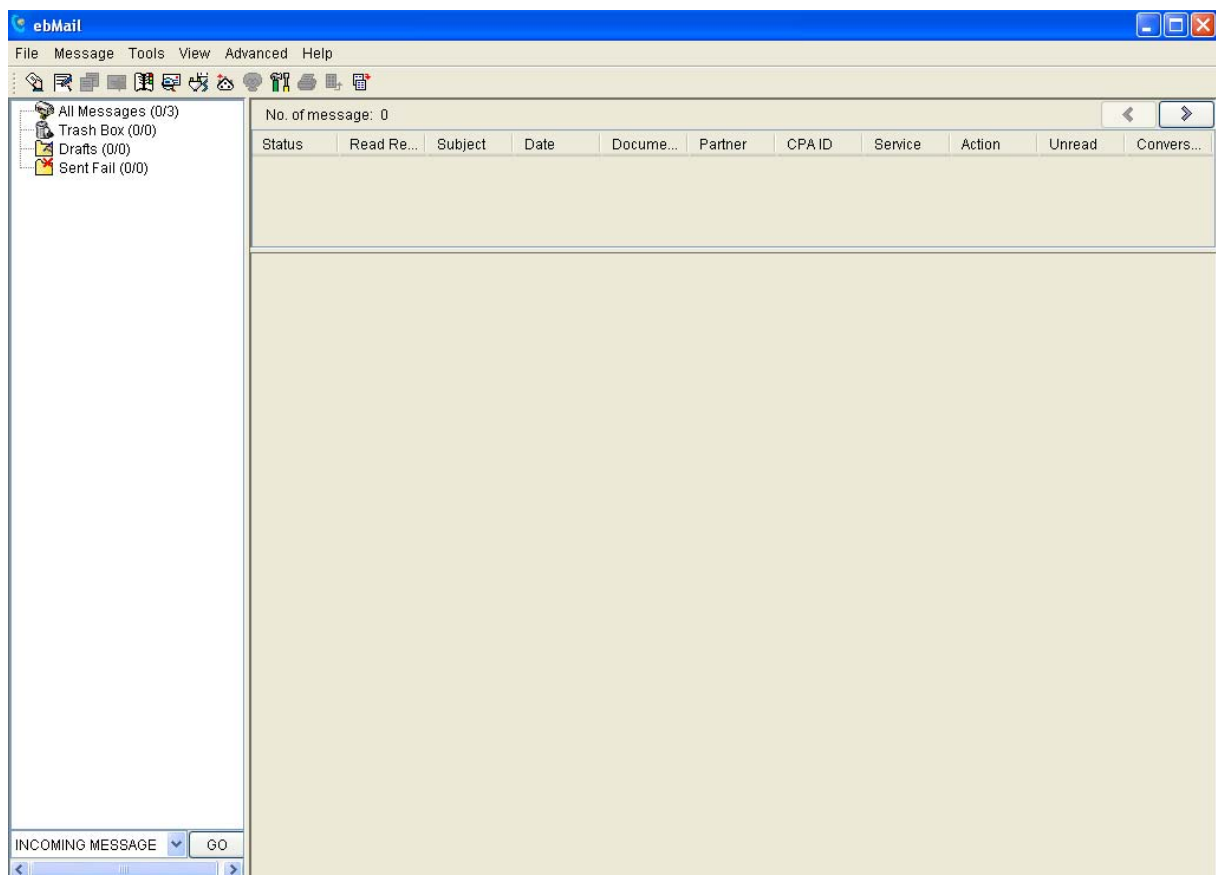


Figura 41 Interfaccia principale di ebMail

Il programma è scritto in java e per quanto riguarda la sua installazione è necessario aver installato sul proprio computer ANT.

Apache Ant è un build tool, utilizzato per la compilazione e l'installazione (deployment) di applicazioni Java complesse. Applicazioni di questo tipo, comportano l'esecuzione di diversi compiti, quali per esempio la compilazione di sorgenti, la creazione e cancellazione di directory di supporto e l'installazione dei vari componenti sviluppati su Server remoti. Ant permette di organizzare, parametrizzare ed automatizzare questo tipo di operazioni.

Dopo aver scaricato e copiato i file binari in una directory sul proprio calcolatore, per il primo utilizzo è necessario compilare i file java, cosa che è facilmente eseguibile portandosi nella cartella in cui si trova ebMail e digitare da finestra di comando “ANT COMPILE”.

Successivamente, se la compilazione è andata a buon fine, ogni volta che si desidera utilizzare il programma, è necessario portarsi nella cartella di ebMail e da finestra di comando digitare “ANT RUN”.

Il programma offre solamente l’infrastruttura per la comunicazione e necessita di alcuni interventi per poter effettivamente implementare una transazione tra due controparti.

Ogni specifica transazione viene infatti implementata come plug-in per l’applicazione base, e deve venir effettivamente implementata secondo uno schema ben definito.

Questi plug-in per poter essere poi utilizzati devono essere caricati nel programma attraverso la specifica procedura visibile in figura:

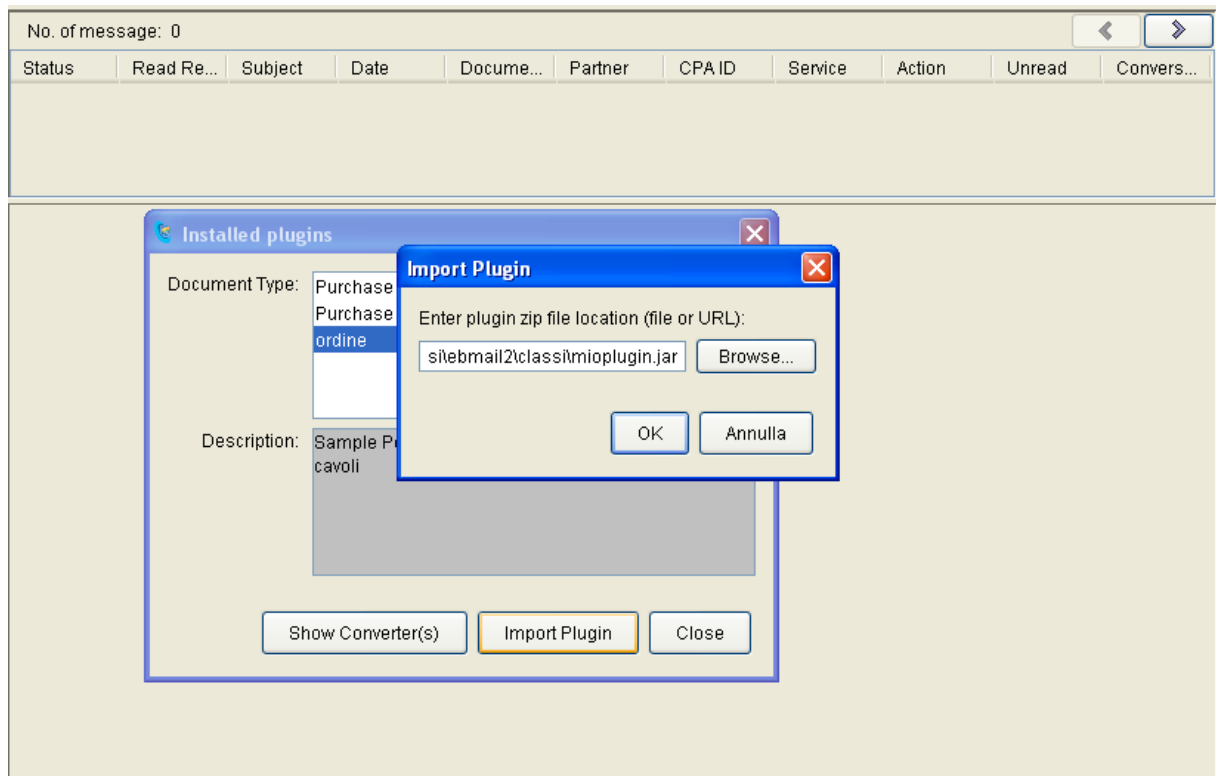


Figura 42 Caricamento di un Plug-in

La progettazione e la realizzazione di questi plug-in rappresentano l’unica difficoltà necessaria per intraprendere transazioni basate su ebXML, scoglio facilmente superabile (anche economicamente) rivolgendosi ad aziende competenti del settore informatico.

Una volta lanciato il programma questo necessita, oltre a caricare il plug-in, che gli vengano forniti i dati relativi all’account di posta elettronica che si desidera utilizzare, quali nome utente, password, sever di posta in ingresso, server di posta in uscita, ecc...

Configure Settings

Account Information

- Partner ID Settings
- Network Settings
- Security
- Plugin

User Information

User name:

E-mail address:

Incoming Mail Server

Server name:

Login name:

Password:

Type: ▼

Port:

Outgoing Mail Server

Server name:

☐ This Outgoing Server requires authentication

Login name:

Password:

OK Annulla

Figura 43 Configurazione del Software

A questo punto è possibile creare un nuovo messaggio come avviene con un normale client di posta elettronica, specificando l'indirizzo del destinatario e l'oggetto del messaggio, con l'unica differenza che invece del testo del messaggio, è possibile comporre il messaggio ebXML tramite l'interfaccia grafica fornita dal plug-in scelto all'inizio della creazione.

Una volta creato, il messaggio viene inviato come se fosse una normalissima e-mail.

Questo fatto rappresenta un vantaggio per gli utenti che non vengono intimiditi da interfacce complicatissime, ma si trovano a che fare con un programma molto simile a quello di un client di posta elettronica che ormai tutti, e molto più chi opera del business, conosce bene.

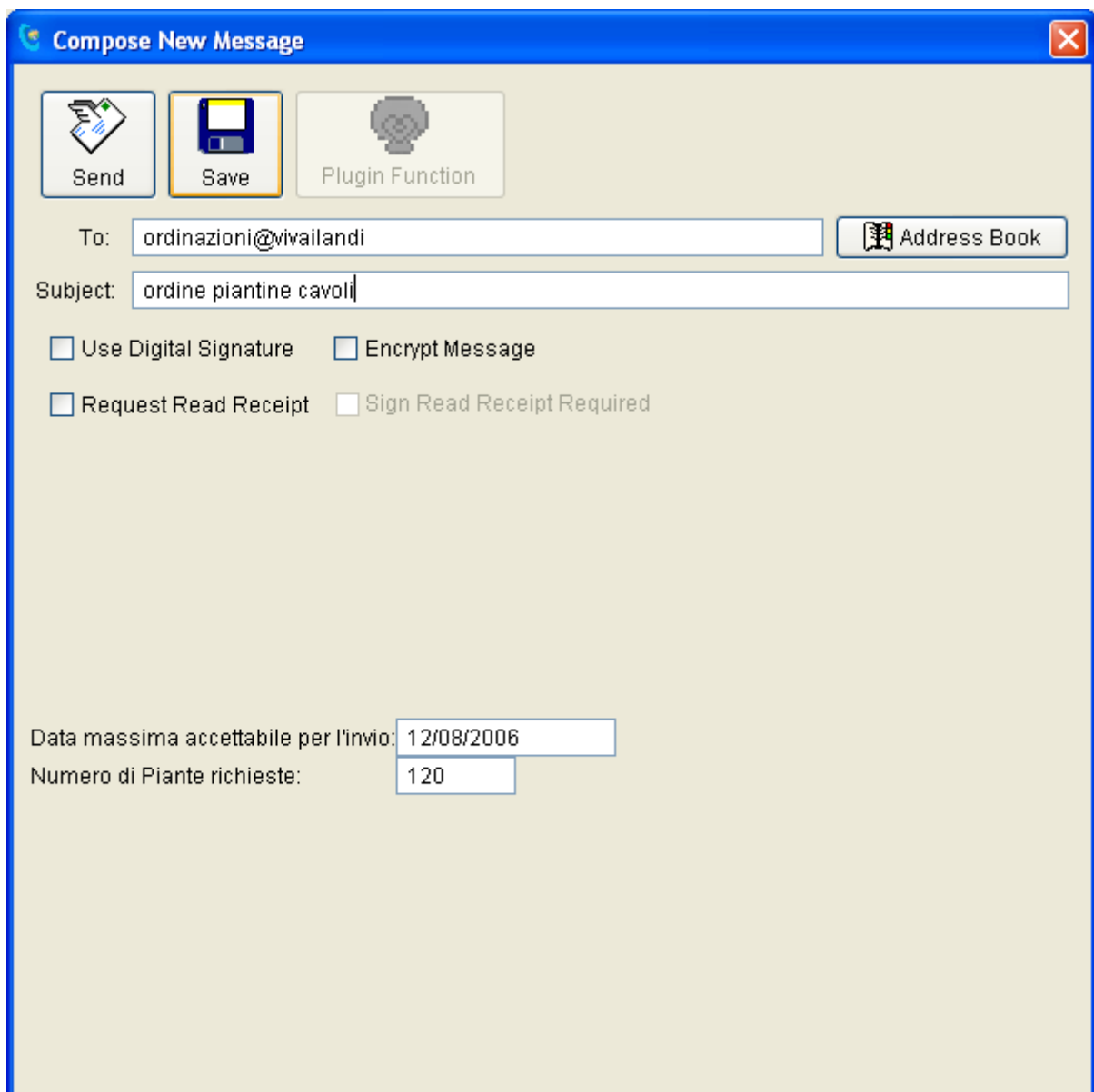


Figura 44 Invio di un messaggio

La ricezione dei messaggi non presenta anch'essa alcuna differenza dalla ricezione di posta elettronica se non la visualizzazione che è affidata ancora al plug-in.

Inoltre, dove previsto dal plug-in, è possibile rispondere in maniera adeguata ai messaggi ebXML, ad esempio con un ack.

3.3.2 Plug-In per ebMail

Il plug-in per ebMail non è altro che un *Java Archive*(JAR), ossia un file compresso contenente alcuni file di classe Java ed un file di configurazione in XML.

Il file XML, che si deve chiamare `PLUGIN.XML` descrive alcuni parametri relativi al plug-in stesso, quali il nome, una descrizione, oltre ovviamente ad un ID univoco all'interno dello

spazio dei nomi dei plug-in, dei parametri relativi alla transazione che vogliamo implementare, tra i quali il nome del CPA a cui facciamo riferimento, al servizio ed all'azione realizzati dal plug-in.

Inoltre vengono specificati i nomi delle classi principali oltre alle azioni relative ai messaggi supportate.

Una classe necessaria all'interno del file jar è quella che si occupa della visualizzazione del plug-in all'interno del programma principale.

Questa classe estende la classe SWING JPanel e si occupa della creazione dell'interfaccia utente utilizzando i componenti tipici della programmazione grafica in java.

È presente un metodo che rende o meno editabile l'interfaccia a seconda che ci troviamo in fase di composizione o di visualizzazione di un messaggio, oltre ai metodi relativi alla lettura e scrittura dei parametri immessi dall'utente.

Altra classe necessaria è quella relativa alla rappresentazione sotto forma di struttura dei dati relativi alla transazione.

Questa è responsabile della memorizzazione dei dati attuali, del caricamento di questi ultimi con dati provenienti da un messaggio salvato o da un messaggio ebXML in ricezione, oltre ovviamente alla creazione del messaggio ebXML partendo dai dati attuali.

È presente anche un metodo attraverso il quale ebMail verifica il corretto inserimento dei dati nella struttura prima di effettuare l'invio del messaggio.

Altra classe contenuta nel file jar è la classe controller, che rappresenta il ponte di comunicazione tra l'interfaccia grafica del plug-in e la classe relativa ai dati.

Essa offre dei metodi per tenere i dati tra le due classi consistenti, oltre a offrire la gestione delle eventuali proprietà o delle eventuali funzioni aggiuntive, quali ad esempio il salvataggio ed il caricamento dei messaggi.

L'ultima classe necessaria presente nel file del plug-in è la classe DocumentHandler che implementa i metodi *ComposeDocument*, *BrowseDocument*, *FollowUpDocument*, responsabili rispettivamente della creazione, della visualizzazione e, ove previsto, della risposta di un messaggio.

All'interno del file jar si possono trovare anche altri file utili per le funzionalità della transazione ma è necessario che vi si trovino i file precedentemente descritti.

In appendice è riportato il codice sorgente delle classi scritte per realizzare un semplice plug-in di esempio relativo all'ordine di piantine di cavoli.

Capitolo 4

Presentazione del prototipo di supporto alla tracciabilità

Ecco in breve, il prototipo di sistema informatico Cerere, preso a riferimento per la tracciabilità di filiera. L'elaborato si concentra sulla funzionalità e sui servizi offerti, tralasciando la parte tecnica.

4.1 Il prototipo Cerere

Il prototipo Cerere nasce come sistema informatico per la tracciabilità nelle filiere agroalimentari.

Come già visto precedentemente il sistema si basa su di un database centralizzato che registra le informazioni provenienti dalla filiera.

Gli utenti principali del sistema dunque risultano facilmente identificati nelle aziende che compongono la filiera e nel nodo di coordinamento.

Altri utenti del sistema possono essere identificati nel consumatore finale, nell'organismo di certificazione e nelle autorità preposte al controllo, che, ognuno con le proprie modalità, può volere accedere alle informazioni registrate dal coordinatore.

Risulta ovvio che tutti gli utenti devono possedere un sistema informatico e la possibilità di accesso ad internet per potere usufruire dei servizi di Cerere.

4.2 Utente Azienda

Le aziende rappresentano l'elemento critico del sistema poiché sono le responsabili della generazione dei dati riguardanti i propri lotti, e sono quindi responsabili della correttezza di questi ultimi e della tempestività con la quale li comunicano.

L'applicazione per le aziende si personalizza all'avvio tramite un file xml ed offre sulla sinistra un pannello con quattro radio button che permettono la selezione del messaggio da inviare.

Il pannello centrale offre dei campi in cui inserire le informazioni principali, quali il codice lotto, la data, i siti, ecc attraverso immissione da tastiera o tramite selezione da Combobox.

sono presenti inoltre dei pulsanti che permettono di accedere a finestre supplementari per specificare proprietà del lotto o della trasformazione, oltre ovviamente al pulsante per l'invio del messaggio.

The screenshot shows the 'TRACE CLIENT' application window. At the top, there's a header with the 'Trace Client' logo on the left, 'Vivai Landi' in the center, and the 'ebXML' logo on the right. Below the header, the main window is titled 'Acquisizione'. On the left side of the main window, there's a vertical menu with four radio buttons: 'Cessione', 'Trasformazione', 'Acquisizione' (which is selected), and 'Movimentazione'. The central area of the 'Acquisizione' window contains several input fields: 'CODICE LOTTO' with the value 'AN433221', 'DATA ACQUISIZIONE' with '2006/04/19 12:45:42', 'SITO DI DESTINAZIONE' with a dropdown menu showing 'Magazzino sementi (MV...', and 'FORNITORE' with a dropdown menu showing 'NATURA (00000000)'. Below these fields are three buttons: 'INSERISCI PROPRIETA' LOTTO', 'INSERISCI PROPRIETA' TRASFORMAZIONE', and 'INVIA MESSAGGIO'. To the right of the main input area, there's a sub-window titled 'Proprietà'. This sub-window has a section 'PROPRIETA' LOTTO' containing an 'Elenco Misurazioni Inserite' table with one entry: '8.4 Kg (Peso)'. There are 'Cancella' and 'AGGIUNGI MISURAZIONE' buttons. Below this, there are dropdown menus for 'Semantica' (set to 'Peso'), 'Unità' (set to 'Kg'), and a text field for 'Misurazione'. At the bottom of the 'Proprietà' sub-window, there's a 'Descrizione' field with 'Sementi' and a 'Messaggio di servizio' text area with an 'Inserisci' button.

Figura 45 Interfaccia per l'invio di un messaggio di Acquisizione

Per l'acquisizione è necessario digitare il codice del lotto mentre la data viene inserita automaticamente, anche se può essere modificata.

Il sito di destinazione e il fornitore sono immessi prelevandoli da un elenco precompilato con i valori presenti nel file di configurazione.

Questa soluzione permette di diminuire in maniera drastica gli errori di digitazione anche se rende un po' più complicata l'aggiunta di nuovi valori.

Per quanto riguarda le proprietà, è possibile scegliere tramite il campo semantica la tipologia di misurazione da inserire, una volta effettuata la scelta, il campo unità si aggiorna automaticamente (il menu di selezione infatti non è abilitato).

Il valore della misurazione viene inserito all'interno del campo Misurazione.

Premendo il pulsante la misurazione viene aggiunta all'elenco delle misurazioni inserite.

La creazione di un messaggio di trasformazione prevede, come in precedenza, l'immissione del codice del lotto generato e della data di generazione.

E' necessario inserire il sito di destinazione ed il sito di trasformazione ed è possibile specificare le proprietà del lotto o della trasformazione.

The screenshot shows the 'TRACE CLIENT' application window. The title bar reads 'TRACE CLIENT'. The interface has a header with the 'Trace Client' logo on the left, 'Vivai Landi' in the center, and the 'ebXML' logo on the right. A left sidebar contains four radio buttons: 'Cessione', 'Trasformazione' (which is selected), 'Acquisizione', and 'Movimentazione'. The main area is divided into two panels. The left panel, titled 'Trasformazione', contains four text input fields: 'CODICE LOTTO GENERATO' (filled with 'AB56328'), 'DATA GENERAZIONE' (filled with '2006/04/19 12:49:39'), 'SITO DI DESTINAZIONE' (a dropdown menu showing 'Serra A (MVL02)'), and 'SITO DI TRASFORMAZIONE' (a dropdown menu showing 'Terreno (MVL01)'). Below these fields are three buttons: 'INSERISCI PROPRIETA' LOTTO', 'INSERISCI PROPRIETA' TRASFORMAZIONE', and 'INSERISCI COMPONENT'. At the bottom of this panel is a large 'INVIA MESSAGGIO' button. The right panel, titled 'Proprieta'', contains a sub-section 'PROPRIETA' LOTTO'. It has a label 'Elenco Misurazioni Inserite' above a dropdown menu showing '100 (Numero di piantine p...)' and a 'Cancella' button. Below this are three more dropdown menus: 'Semantica' (showing 'Numero di piantine presen...'), 'Unita', and 'Misurazione'. An 'AGGIUNGI MISURAZIONE' button is positioned below the 'Misurazione' dropdown. At the bottom of the 'Proprieta'' panel, there is a 'Descrizione' label above a text input field containing 'Cassa di piantine', and a 'Messaggio di servizio' label above a large empty text area. An 'Inserisci' button is located at the bottom left of the 'Proprieta'' panel.

Figura 46 Interfaccia per l'invio di un messaggio di Trasformazione

Risulta inoltre obbligatorio l'inserimento di almeno un altro campo, e cioè il lotto, o i lotti, che sono ingresso alla trasformazione.

L'immissione avviene tramite l'apposito bottone che apre il relativo pannello, ed è simile all'inserimento di una proprietà.

E' possibile aggiungere caratteristiche ad ogni lotto componente.

Rispetto al prototipo originale, durante lo svolgimento della tesi, è stata aggiunta una nuova interfaccia per la movimentazione dei lotti sotto la supervisione dello stesso attore responsabile.

Figura 47 interfaccia per l'invio di un messaggio d Movimentazione

Si tratta semplicemente di una semplificazione per l'utente, per rendere più intuitivo lo spostamento da un sito all'altro, in quanto il messaggio viene mappato sopra un messaggio di trasformazione.

Anche l'interfaccia è del tutto identica, a parte i nomi dei campi e dei bottoni, ad un'interfaccia di trasformazione.

L'interfaccia di cessione è la più semplice di tutte, non potendo specificare nessuna proprietà per la cessione o per il lotto.

The screenshot shows the TRACE CLIENT application window. The title bar reads "TRACE CLIENT". The interface has a header bar with the "Trace Client" logo on the left, the text "Vivai Landi" in the center, and the "ebXML" logo on the right. Below the header, there is a sub-header bar labeled "Cessione". On the left side, there is a vertical menu with four radio buttons: "Cessione" (selected), "Trasformazione", "Acquisizione", and "Movimentazione". The main area contains a form with four fields: "CODICE LOTTO" with the value "FD555377", "DATA CESSIONE" with the value "2006/04/19 12:52:59", "SITO SORGENTE" with a dropdown menu showing "Serra A (MVL02)", and "CLIENTE" with a dropdown menu showing "NATURA (00000000)". Below these fields is a button labeled "INVIA MESSAGGIO".

Figura 48 Interfaccia per l'invio di un messaggio di Cessione

E' necessario l'immissione del lotto da cedere e della data di cessione, oltre alla selezione tramite menù del sito sorgente e dell'attore responsabile a cui viene ceduto il lotto.

Ogni volta che viene premuto il tasto di invio messaggio, viene chiesta una conferma dell'operazione e successivamente le informazioni estratte vanno a comporre il messaggio xml che verrà inviato al sistema di messaggistica.

4.3 Utente Coordinatore

Il nodo di coordinamento come già detto, è il responsabile della memorizzazione dei messaggi inviati dalle Aziende.

Dunque il sistema a lui preposto deve permettere il continuo monitoraggio dei flussi dei dati provenienti dalle Aziende.



The screenshot shows a window titled "COORDINATORE" with a menu bar containing "MESSAGGI RICEVUTI (16)", "STATISTICHE", and "CAMBI RESPONSABILITA'". The main area displays a table with the following data:

| Content ID | Data Ricezione | Tipo Messaggio | Attore | Codice Lotto |
|------------|---------------------|----------------|--------|--------------|
| 0 | 2004/07/06 18:45:28 | Acquisizione | 1234 | LC1 |
| 1 | 2004/07/06 18:45:28 | Acquisizione | 1234 | LC2 |
| 2 | 2004/07/06 18:45:28 | Cessione | 1234 | LC1 |
| 3 | 2004/07/06 18:45:28 | Acquisizione | 5678 | LC1 |
| 4 | 2004/07/06 18:45:29 | Acquisizione | 5678 | LC2 |
| 5 | 2004/07/06 18:45:29 | Cessione | 1234 | LC2 |
| 6 | 2004/07/06 18:45:30 | Trasformazione | 5678 | LC3 |
| 7 | 2004/07/06 18:45:30 | Cessione | 5678 | LC3 |
| 8 | 2004/07/06 18:45:31 | Acquisizione | 9999 | LC3 |
| 9 | 2004/07/06 18:45:31 | Trasformazione | 9999 | LC4 |
| 10 | 2004/07/06 18:45:31 | Trasformazione | 9999 | LC5 |
| 11 | 2004/07/06 18:45:44 | Trasformazione | 9999 | LC6 |
| 12 | 2004/07/06 18:45:45 | Trasformazione | 9999 | LC7 |
| 13 | 2004/07/06 18:45:45 | Trasformazione | 9999 | LC8 |
| 14 | 2004/07/06 18:45:58 | Trasformazione | 9999 | LC9 |
| 15 | 2004/07/06 18:45:59 | Trasformazione | 9999 | LC10 |

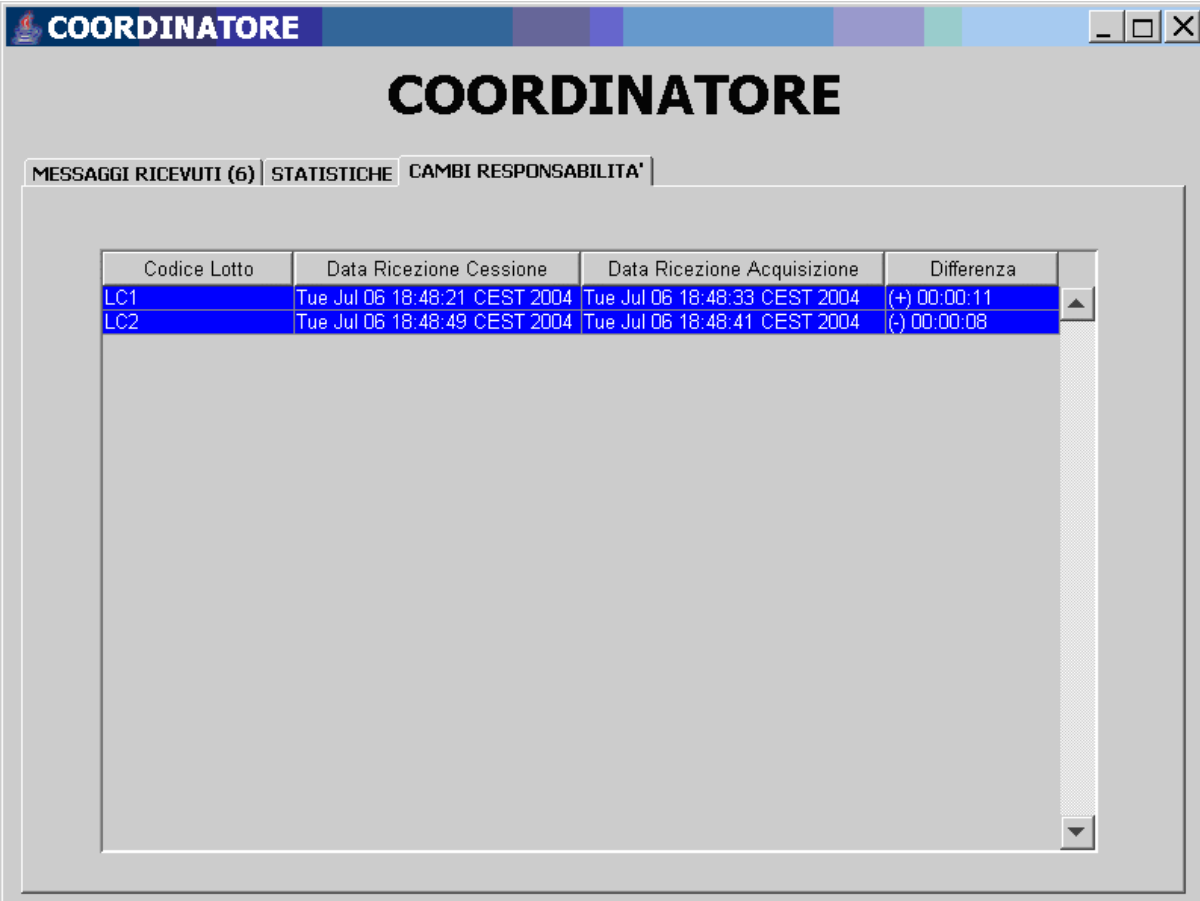
Figura 49 Interfaccia per nodo di coordinamento

L'applicazione presenta tre sezioni di cui la prima, la più importante, mostra le informazioni sintetiche sui MESSAGGI RICEVUTI.

Queste informazioni sono l'ID assegnato dal sistema al messaggio, la data e l'ora di ricezione, il tipo di messaggio (acquisizione, trasformazione, cessione), il codice dell'attore che ha inviato il messaggio ed infine il codice del lotto riguardante il messaggio.

Il secondo pannello riporta delle semplici STATISTICHE riassuntive sul numero totale dei messaggi e il numero dei messaggi ricevuti e numero di Byte ricevuti.

L'ultima sezione, anch'essa molto importante, risulta essere relativa ai cambi di responsabilità. Questo registra, ogni volta che riceve un messaggio di acquisizione/cessione le informazioni relative, e le completa indicando anche il tempo trascorso, quando arriva il corrispettivo messaggio di cessione/acquisizione.



The screenshot shows a software window titled "COORDINATORE". Inside, there are three tabs: "MESSAGGI RICEVUTI (6)", "STATISTICHE", and "CAMBI RESPONSABILITA'". The "CAMBI RESPONSABILITA'" tab is active, displaying a table with the following data:

| Codice Lotto | Data Ricezione Cessione | Data Ricezione Acquisizione | Differenza |
|--------------|-------------------------------|-------------------------------|--------------|
| LC1 | Tue Jul 06 18:48:21 CEST 2004 | Tue Jul 06 18:48:33 CEST 2004 | (+) 00:00:11 |
| LC2 | Tue Jul 06 18:48:49 CEST 2004 | Tue Jul 06 18:48:41 CEST 2004 | (-) 00:00:08 |

Figura 50 Registrazione dei cambi di responsabilità

Come si vede nella figura il campo differenza possiede un valore con segno : questo valore è positivo nel caso in cui arrivi prima il messaggio di cessione, negativo nel caso contrario

4.4 Altri utenti

Gli altri utenti, che ricordiamo sono i consumatori, le Autorità sanitarie o gli organismi di certificazione, possono accedere al sistema tramite il portale WEB.

Attraverso il sito gli utenti possono compiere ricerche ed analisi accurate dei flussi.

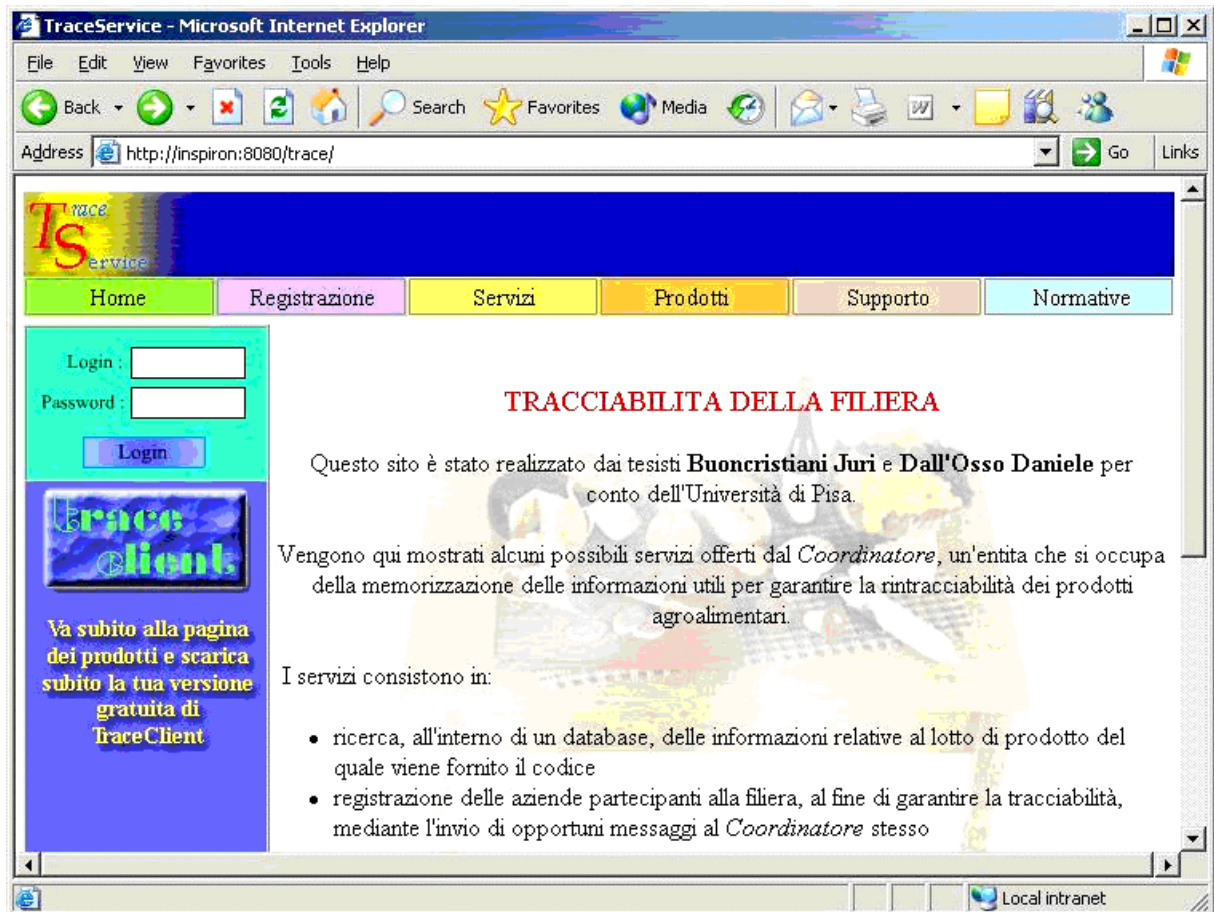


Figura 51 Home page del portale WEB

Nella pagina principale, sulla destra, sono presenti i campi per il login degli utenti preventivamente registrati. Una volta autenticati è possibile accedere alle varie sezioni del portale. Non tutte sono ancora pienamente funzionanti.

La sezione fondamentale è quella dei SERVIZI dalla quale si interroga il sistema inserendo il codice di un lotto



Figura 52 Pagina per la ricerca di un lotto

La figura successiva mostra un esempio di una schermata derivante dalla ricerca di un lotto di pasta

Processi e relativi lotti lavorati

Lotti da cui ha origine il lotto selezionato tramite i processi di lavorazione sotto riportati. Cliccare su un lotto per risalire ai passaggi precedenti della filiera.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|-------------------------------|------------------------|---------------|-------------|----------------------------------|--------------------------|
| LGD2-76898709 | 0024-10-24 16:50:00 | Grano Duro | Macinazione | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LGD2-76898710 | 0029-10-25 13:40:00 | Grano Duro | Macinazione | Azienda Agricola La Spiga s.n.c. | Dettagli |

Lotto selezionato

Informazioni sul lotto selezionato.



Codice Lotto : **LFA2-I243443**
 Data Generazione : **0006-11-25 09:25:00**
 Nome Prodotto : **Farina**
 Locazione : **Deposito Via Tanaro, 107**
 Azienda Responsabile : **Azienda Agricola La Spiga s.n.c.**
 Peso : **400.0 Kg**
 Proteine : **11.0 %**
 Glutine : **16.0 %**
 Forza : **300.0 W**
 Assorbimento : **59.0 %**

Processi e relativi lotti prodotti in uscita

Prodotti in cui è stato trasformato il lotto selezionato.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|------------------------------|------------------------|-------------------------------------|----------------------------------|----------------------------------|--------------------------|
| LFF2-1485900 | 0007-10-25 08:55:00 | Confezioni di farina tipo 00 da 1Kg | Confezionamento in pacchi da 1Kg | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LFF2-1485901 | 0007-10-25 09:05:00 | Confezioni di farina tipo 00 da 1Kg | Confezionamento in pacchi da 1Kg | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LFF2-1485902 | 0007-10-25 09:15:00 | Confezioni di farina tipo 00 da 1Kg | Confezionamento in pacchi da 1Kg | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LSF2-7987352 | 0007-10-25 08:05:00 | Sacco di Farina | Confezionamento in sacchi | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LSF2-7987353 | 0007-10-25 08:15:00 | Sacco di Farina | Confezionamento in sacchi | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LSF2-7987354 | 0007-10-25 08:25:00 | Sacco di Farina | Confezionamento in sacchi | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LSF2-7987355 | 0007-10-25 08:35:00 | Sacco di Farina | Confezionamento in sacchi | Azienda Agricola La Spiga s.n.c. | Dettagli |
| LSF2-7987356 | 0007-10-25 08:45:00 | Sacco di Farina | Confezionamento in sacchi | Azienda Agricola La Spiga s.n.c. | Dettagli |

Figura 53 Risultato di una ricerca

La schermata è essenziale: sono stati preferiti gli aspetti funzionali rispetto all'estetica, garantendo al cliente interfaccia di facile comprensione ed utilizzo.

Nella sezione centrale; LOTTO SELEZIONATO, sono disponibili in forma di scheda i dati riassuntivi del lotto.

Grande vantaggio offerto dal sistema quello di rendere disponibili le informazioni dei lotti correlati a quello della ricerca, relativi a un step precedente e successivo nella catena di tracciatura.

Sopra la scheda del lotto, nella sezione processi e relativi lotti lavorati, sono presenti i lotti da cui ha avuto origine il codice selezionato, mentre in basso sono presenti, nella sezione processi e relativi lotti prodotti in uscita, tutti i lotti che in qualche modo derivano da quello selezionato.

La presenza di collegamenti ipertestuali, in corrispondenza dei codici dei lotti, permette un facile e chiaro spostamento tra i lotti, permettendo così di tracciare un lotto, di ricostruire cioè il suo percorso dalla natura al consumatore finale.

Capitolo 5

Validazione del sistema informatico

Questo capitolo, conclusivo della tesi, prevede la validazione del sistema di tracciabilità Cerere sul contesto degli ortaggi. La simulazione di una filiera prototipo permette la verifica della parte software, ma è utile per mettere in luce carenze anche nel modello dei dati.

5.1 Finalità della validazione

Una parte della presente tesi è quella di validare il software sulla filiera degli ortaggi ed al contempo, il controllo della modellazione stessa.

Con il termine di validazione si intende una verifica della rispondenza ai requisiti del modello soprattutto della piattaforma infomatica.

L'obiettivo della validazione è una valutazione degli aspetti funzionali del sistema, allo scopo di scoprire eventuali errori o carenze nel software ed all'individuazione di eventuali migliorie. Gli aspetti fondamentali su cui è stata focalizzata l'attenzione durante il processo di validazione sono stati l'efficacia del sistema di tracciatura dei lotti, intesa come la capacità di inserire tutte le caratteristiche emerse dal modello, e la correttezza e validità dei dati ottenuti dalle interrogazioni al portale.

Per partire con la validazione è stato necessario creare un set di dati plausibili, estrapolati dal modello ricavato in precedenza.

Gli attori principali sono quattro:

- Vivai Landi
- Azienda Agricola Maestràle
- Rossini Confezionamenti
- Bianchi Distribuzioni

Più i due attori fittizi necessari per l'inizio e la fine della filiera:

- Natura

- Consumatore

Come passo successivo sono stati creati i file di configurazione per i client delle aziende, riportati per intero in appendice.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Document      : config.xml
  Created on    : 27 Febbraio 2006, 20.37
  Author       : Marco
  Description: file di configurazione per traceClient Landi vivai
-->
<config code="VL00001" name="Vivai Landi">
  <ActorList>
    <Actor code="0000000" name="NATURA" type="supplier"/>
    <Actor code="AM00002" name="Azienda agricola Maestrale"
type="customer"/>
  </ActorList>
  <SiteList>
    <Site code="MVL02" name="Serra A"/>
    <Site code="MVL04" name="Magazzino sementi"/>
    <Site code="MVL01" name="Terreno"/>
    <Site code="MVL03" name="Serra B"/>
    <Site code="MVL05" name="Furgone"/>
  </SiteList>
  <MeasureTypeList>
    <Measure unit="Kg" semantic="Peso"/>
    <Measure unit="" semantic="Numero di piantine presenti nel
contenitore"/>
  </MeasureTypeList>
</config>
```

Attraverso questo file vengono inizializzati i combo box relativi alle misurazioni (unità e semantica) , quelli relativi ai siti (codice e nome del sito) e quelli relativi agli attori responsabili (codice e tipo:fornitore, Acquirente, Fornitore/Acquirente)

Inoltre, affinché il sistema registri i messaggi correttamente, è necessario aggiornare il DataBase sul lato coordinatore, inserendo le informazioni relative ai codici degli attori, dei siti e delle misurazioni.

Tutto ciò viene effettuato per mezzo dello script SQL riportato in appendice A.

A questo punto è possibile partire con l'immissione dei dati tramite il software.

5.2 I Pacchetti Informativi

L'immissione dei dati e la creazione dei messaggi risulta molto semplice grazie alla presenza delle interfacce grafiche, prestandosi facilmente anche all'uso da parte dei meno esperti.

La figura illustra un esempio di codifica nello standard ebXML di un messaggio di acquisizione.

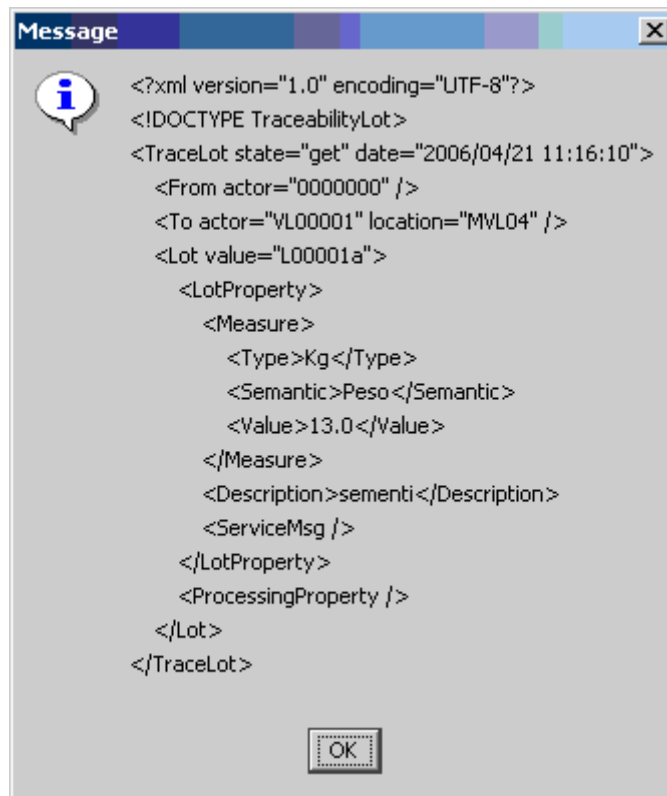


Figura 54 Codifica ebXML di un messaggio di acquisizione.

All'interno del messaggio sono presenti il tipo di quest'ultimo (*TraceLot state*) che in questo caso, trattandosi di un messaggio di acquisizione, è di tipo "get", la data, il codice dell'attore che fornisce il lotto (*From actor*), il codice dell'attore che acquisisce il lotto (*To actor*), il codice del sito di destinazione (*location*), il codice del lotto generato (*Lot value*) e le eventuali proprietà di lotto (*LotProperty*) e di processo (*ProcessingProperty*).

In questo caso è presente una misurazione del peso delle sementi del valore di 13 Kg, oltre alla descrizione del lotto stesso.

La figura seguente mostra, invece, un esempio di codifica ebXML di un messaggio di trasformazione.

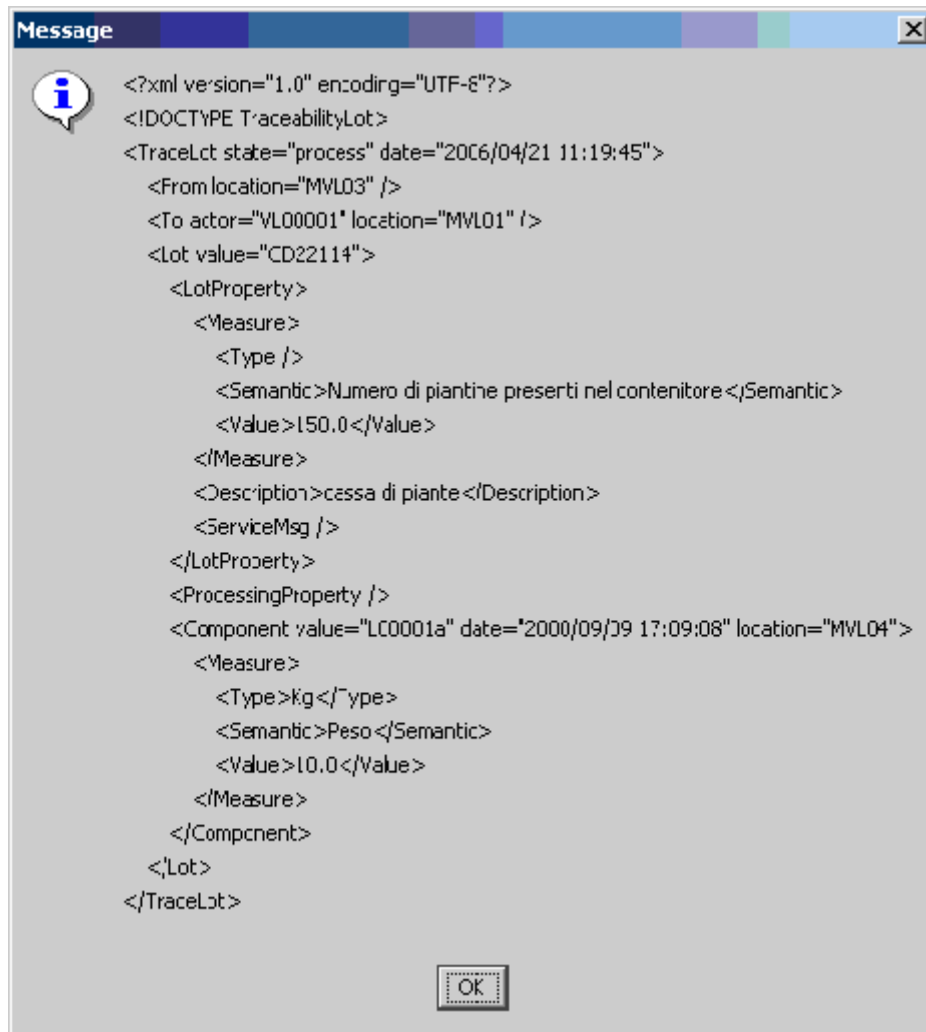


Figura 55 Codifica ebXML di un messaggio di trasformazione.

Ancora una volta abbiamo il tipo di messaggio, che in questo caso è “*process*”, la data di generazione, il sito di trasformazione, l’attore responsabile, il sito di destinazione ed il codice del lotto generato.

In questo esempio è presente una misurazione relativa al numero di piante presenti nelle casse alveolari; come si nota il campo *type* è aperto e chiuso subito, non essendo presente un unità di misura per questa misurazione.

All’interno del campo *processing property* vengono elencati i componenti (*Component*) della trasformazione.

Per ciascuno di questi viene indicato ovviamente il codice (*value*), la data (di immissione) ed il sito di provenienza (*location*).

Nel nostro caso è stata anche specificata una misurazione relativa al lotto componente.

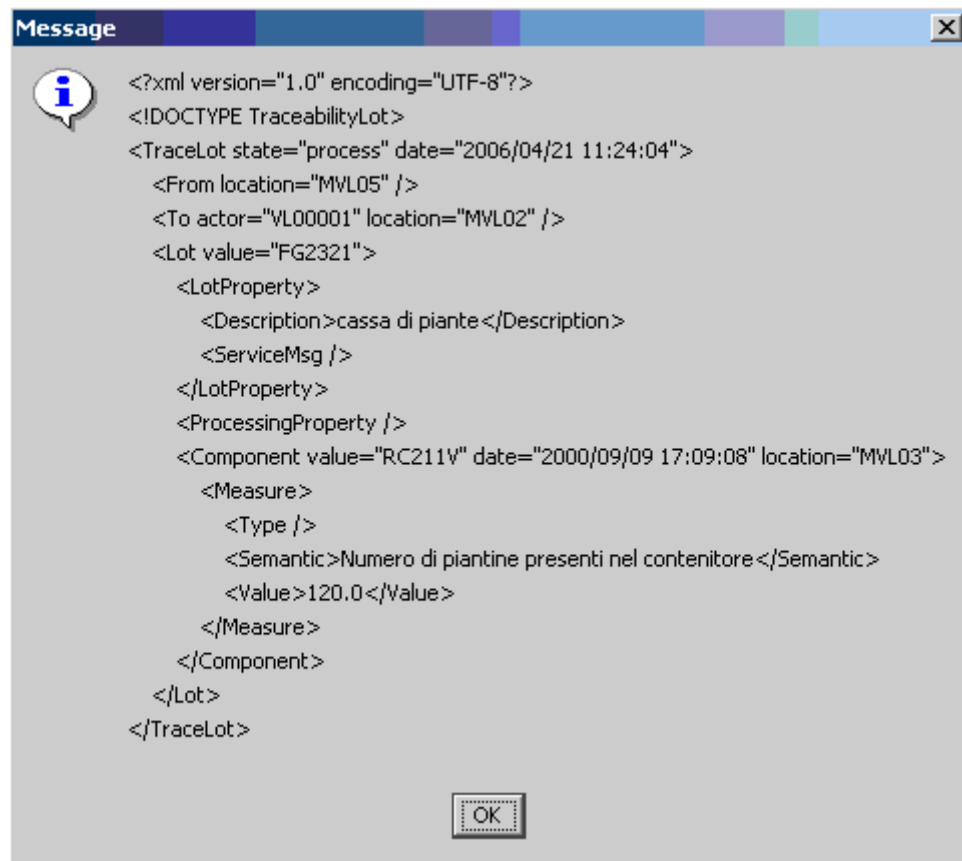


Figura 56 Codifica ebXML di un messaggio di Movimentazione

Come già accennato in precedenza la movimentazione viene mappata su un messaggio di trasformazione, in quanto il sistema, dal lato coordinatore, riconosce solo tre tipi di messaggio: acquisizione, cessione e trasformazione.

Il messaggio non introduce nessun nuovo aspetto e pertanto possiamo passare ad analizzare l'ultimo tipo.

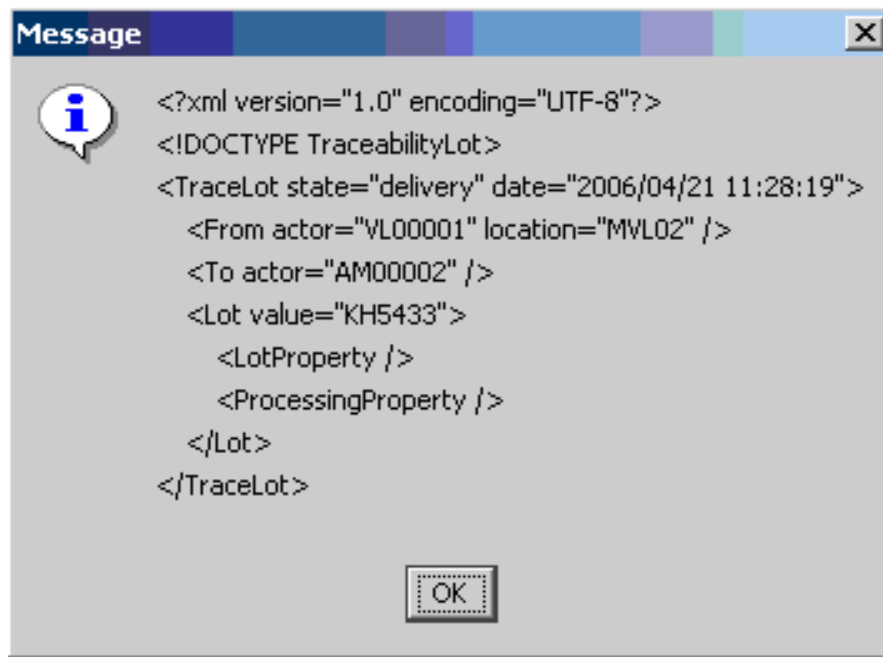


Figura 57 Codifica ebXML di un messaggio di Cessione

Il messaggio è molto semplice in quanto non è possibile inserire nessuna proprietà.

Il tipo del messaggio è “*delivery*” e si specifica la data di cessione, l’attore che cede il lotto, il sito da cui è questo è prelevato per la cessione, l’attore a cui il lotto viene ceduto e, ovviamente, il codice del lotto stesso.

Un'anomalia riscontrata durante la fase di immissione dei dati della filiera riguarda la possibilità di inserire nel sistema caratteristiche categoriche, ovvero la possibilità di inserire del testo nel campo misurazione.

Il sistema, infatti, non accetta ad esempio il valore della misurazione “media” attribuito al parametro “Consistenza” e presenta una finestra con una segnalazione di errore.

Il problema è nel nostro caso facilmente risolvibile utilizzando una traduzione numerica dei relativi valori, anche se questo può facilmente introdurre delle incomprensioni.

5.3 Ricezione dei messaggi.

Il prototipo Cerere si appoggia su ebMS per l’invio dei pacchetti informativi, e più precisamente su Hermes MSH, sviluppato anch’esso sotto la supervisione del CECID e dell’università di Honk Kong. Questo sistema di messaggistica garantisce il transito dei pacchetti, attraverso reti informatiche, dagli attori della filiera al nodo di coordinamento e della loro archiviazione nel database del sistema.

L'applicazione per il coordinatore, una volta avviata, visualizza costantemente la ricezione dei messaggi e lo svolgersi delle attività all'interno della filiera da lui supervisionata.

La figura seguente mostra l'arrivo di alcuni pacchetti di prova inviati per svolgere test sull'applicazione .

| Content ID | Data Ricezione | Tipo Messaggio | Attore | Codice Lotto |
|-------------------------|---------------------|----------------|---------|--------------|
| 20060608-103145160-C... | 2006/06/08 12:32:10 | Acquisizione | AM00002 | VL004 |
| 20060608-103630861-C... | 2006/06/08 12:36:56 | Trasformazione | AM00002 | AM001 |
| 20060608-103805467-C... | 2006/06/08 12:38:30 | Trasformazione | AM00002 | AM002 |
| 20060608-104002885-C... | 2006/06/08 12:40:27 | Trasformazione | AM00002 | AM003 |
| 20060608-104105496-C... | 2006/06/08 12:41:31 | Cessione | AM00002 | AM003 |
| 20060608-104207745-C... | 2006/06/08 12:42:34 | Acquisizione | RC00002 | AM003 |
| 20060608-104450239-C... | 2006/06/08 12:45:17 | Trasformazione | RC00002 | RC001 |
| 20060608-104605567-C... | 2006/06/08 12:46:32 | Trasformazione | RC00002 | RC002 |
| 20060608-104746963-C... | 2006/06/08 12:48:12 | Trasformazione | RC00002 | RC003 |
| 20060608-104924864-C... | 2006/06/08 12:49:51 | Trasformazione | RC00002 | RC004 |
| 20060608-104958352-C... | 2006/06/08 12:50:24 | Cessione | RC00002 | RC004 |
| 20060608-105139217-C... | 2006/06/08 12:52:04 | Acquisizione | BD00001 | |
| 20060608-105304650-C... | 2006/06/08 12:53:31 | Acquisizione | BD00001 | RC004 |
| 20060608-105354051-C... | 2006/06/08 12:54:20 | Trasformazione | BD00001 | BD001 |
| 20060608-105452485-C... | 2006/06/08 12:55:17 | Trasformazione | BD00001 | BD002 |
| 20060608-105523549-C... | 2006/06/08 12:55:50 | Cessione | BD00001 | BD001 |

Figura 58 Finestra di ricezione dei messaggi.

La figura sottostante presenta la sezione del coordinatore relativa ai cambi di responsabilità avvenuti. Affinché la tracciatura avvenga in maniera corretta è necessario che i lotti di cessione e di acquisizione mantengano lo stesso codice identificativo; questo comporta una comunicazione tra i due attori coinvolti nella transazione. L'attribuzione del nuovo codice da parte dell'acquirente può essere eseguita attraverso una successiva trasformazione. Un cambio di responsabilità correttamente rilevato viene visualizzato nella finestra relativa mostrando sia la data di cessione e quella di acquisizione. Dove non sono presenti entrambe le date il cambio di responsabilità è incompleto o non correttamente tracciato.

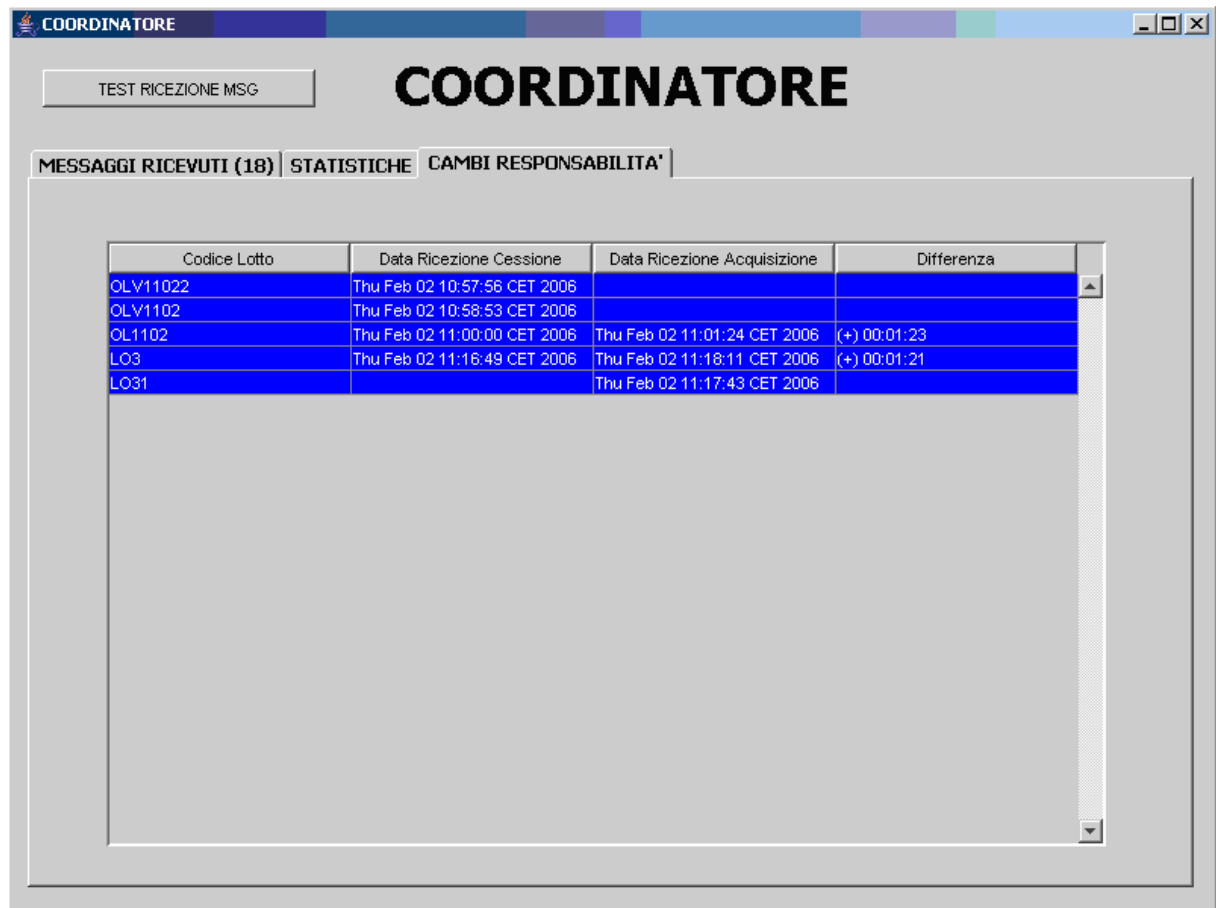


Figura 59 Finestra di visualizzazione dei cambi di responsabilità.

5.4 Verifica dell'avvenuta tracciatura dei lotti

La conclusione della validazione del sistema si concretizza con l'accesso al sito web e l'esecuzione di ricerche mirate sui lotti tracciati. Obiettivo della ricerca è di verificare la completezza e la correttezza delle informazioni riportate dalla piattaforma e la loro consistenza con i dati inseriti. La verifica si avvia con l'immissione di uno qualsiasi dei codici inseriti durante la tracciatura attraverso le applicazioni per le aziende.

The screenshot displays the Trace service web application interface. At the top, there is a blue header with the 'Trace service' logo on the left and a navigation menu on the right with buttons for 'Home', 'Registrazione', 'Servizi', 'Prodotti', 'Supporto', and 'Normative'. Below the header, a green bar reads 'Benvenuto demo'. On the left side, there is a vertical menu with links for 'Cambia Profilo', 'Modifica Password', and 'Logout'. Below this menu is a blue box containing the 'Trace client' logo and a promotional message: 'Va subito alla pagina dei prodotti e scarica subito la tua versione gratuita di TraceClient'. The main content area features the text 'BENVENUTO DEMO' in red, followed by the instruction 'Inserisci il codice del lotto stampato sulla confezione del prodotto.' Below this, there is a form with the label 'Codice Lotto :', a text input field containing 'VL01', and an 'Invia' button. The background of the main content area shows a faint image of a table with various items on it.

Figura 60 Ricerca di un codice.

Il primo codice inserito è relativo al lotto di sementi generato dall'azienda vivai Landi in seguito all'attività di acquisizione da natura. Vediamo nella figura 61 la schermata fornita dal sistema in questo primo caso.

TABELLA DELLA RINTRACCIABILITÀ
LOTTO N° VL01

Lotto selezionato
Informazioni sul lotto selezionato.

| | |
|-------------------------|------------------------------------|
| Codice Lotto : | VL01 |
| Data Generazione : | 2006-05-03 10:46:03 |
| Nome Prodotto : | sementi |
| Messaggio di Servizio : | |
| Locazione : | Magazzino sementi Via Mascagni, 90 |
| Azienda Responsabile : | Vivai Landi |
| Peso : | 12.3 Kg |

Processi e relativi lotti prodotti in uscita
Prodotti in cui è stato trasformato il lotto selezionato.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|----------------------|---------------------|-----------------|----------|-------------|--------------------------|
| VL02 | 2006-05-03 10:49:46 | | | Vivai Landi | Dettagli |
| VL03 | 2006-05-03 10:55:17 | Cassa di piante | | Vivai Landi | Dettagli |

Figura 61 Schermata 1 dei risultati.

Il form *Lotto selezionato* riassume tutte le informazioni relative al lotto oggetto della ricerca, fra le quali è possibile notare la caratteristica numerica peso del valore di 12,3 Kg.

Nella sezione posta al di sotto, *Processi e relativi lotti in uscita*, troviamo alcune informazioni principali, tra le quali è importante notare il fornitore, sui lotti di codice VL02 e VL03, entrambi generati dalla trasformazione del lotto preso in esame.

A seguito dei messaggi pervenuti e analizzati, il sistema è riuscito a tracciare questa relazione che lega tre lotti, uno di input e due di output, ad una trasformazione effettuata sotto la responsabilità dello stesso attore.

La sezione *Processi e relativi lotti in uscita* presenta inoltre dei collegamenti ipertestuali che ci permettono di proseguire nell'esplorazione dei dati tracciati, passando ad esempio alla visualizzazione dei dati relativi al lotto di codice VL03.

La pagina fornisce l'indicazione dei lotti da cui quello selezionato deriva (in questo caso VL01) nella sezione in alto denominata *Processi e relativi lotti lavorati*, oltre ovviamente di quelli prodotti, nella sezione in basso già vista nella prima schermata (figura 62).



The screenshot shows the Trace service web application interface. At the top, there is a navigation bar with links: Home, Registrazione, Servizi, Prodotti, Supporto, and Normative. Below this, a sidebar on the left contains links: Benvenuto demo, Cambia Profilo, Modifica Password, Logout, and a section for TraceClient with a download link. The main content area is titled "TABELLA DELLA RINTRACCIABILITÀ LOTTO N° VL03". It features three main sections: "Processi e relativi lotti lavorati", "Lotto selezionato", and "Processi e relativi lotti prodotti in uscita".

Processi e relativi lotti lavorati
Lotti da cui ha origine il lotto selezionato tramite i processi di lavorazione sotto riportati. Cliccare su un lotto per risalire ai passaggi precedenti della filiera.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|----------------------|---------------------|---------------|----------|-------------|--------------------------|
| VL01 | 2006-05-03 10:46:03 | sementi | | Vivai Landi | Dettagli |

Lotto selezionato
Informazioni sul lotto selezionato.

| | |
|--|--------------------------|
| Codice Lotto : | VL03 |
| Data Generazione : | 2006-05-03 10:55:17 |
| Nome Prodotto : | Cassa di piante |
| Messaggio di Servizio : | |
| Locazione : | Serra A Via Mascagni, 88 |
| Azienda Responsabile : | Vivai Landi |
| Trasformazioni associate al lotto N° VL01 : | |
| Codice Lotto : | VL01 |
| Nome Prodotto : | sementi |
| Trasformazione : | |
| Peso : | 6.0 Kg |

Processi e relativi lotti prodotti in uscita
Prodotti in cui è stato trasformato il lotto selezionato.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|----------------------|---------------------|-----------------|----------|----------------------------|--------------------------|
| VL03 | 2006-05-03 11:00:15 | cassa di piante | | Azienda agricola Maestrale | Dettagli |

Figura 62 Schermata 2 dei risultati.

Ai fini della validazione è stata necessaria l'esplorazione di ogni pagina di collegamento per accertare la presenza di eventuali interruzioni del flusso di dati tracciato.

In particolare è stata prestata molta attenzione all'effettiva registrazione dei cambi di responsabilità.

La schermata informativa sul lotto di codice VLO3, appartenente ancora all'Azienda vivai Landi, mostra l'importanza delle informazioni sintetiche presenti nei lotti prodotti o lavorati, preannunciando una cessione nei confronti dell'Azienda agricola Maestrale ed il relativo cambio di responsabilità (figura 63).

Risulta chiaro adesso che il codice di cessione e di acquisizione deve essere identico, come rilevato in precedenza, onde evitare che il sistema non riesca a rilevare il cambio di responsabilità ed interrompere così la tracciatura dei dati al lotto ceduto.

TABELLA DELLA RINTRACCIABILITÀ
LOTTO N° VLO3

Processi e relativi lotti lavorati
Lotti da cui ha origine il lotto selezionato tramite i processi di lavorazione sotto riportati. Cliccare su un lotto per risalire ai passaggi precedenti della filiera.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|----------------------|---------------------|-----------------|----------|-------------|--------------------------|
| VLO3 | 2006-05-03 10:55:17 | Cassa di piante | | Vivai Landi | Dettagli |

Lotto selezionato
Informazioni sul lotto selezionato.



Codice Lotto : **VL03**

Data Generazione : **2006-05-03 11:00:15**

Nome Prodotto : **cassa di piante**

Messaggio di Servizio :

Locazione : **Terreno A Via Metauro, 12**

Azienda Responsabile : **Azienda agricola Maestrale**

Trasformazioni associate al lotto N° VL03 :

Codice Lotto : **VL03**

Nome Prodotto : **Cassa di piante**

Trasformazione :

Processi e relativi lotti prodotti in uscita
Prodotti in cui è stato trasformato il lotto selezionato.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|----------------------|---------------------|---------------|----------|----------------------------|--------------------------|
| AM02 | 2006-05-03 11:04:22 | | | Azienda agricola Maestrale | Dettagli |

Figura 63 Schermata del primo cambio di responsabilità.

Continuando con questo processo, si arriva all'ultima azienda della filiera, nella quale, per terminare la filiera, è necessario effettuare una cessione al Consumatore, attività che rappresenta la vendita al dettaglio.

Il sistema non ha riportato lo svolgersi del cambio di responsabilità, fermandosi alla tracciatura dell'ultimo lotto prodotto dall'Azienda Bianchi Distribuzioni.

Questo fatto molto probabilmente è dovuto al fatto che il Consumatore non rappresenta un attore reale ma un attore fittizio ed il sistema non riesce a completare la registrazione del cambio di responsabilità.

The screenshot displays the Trace Client web application. The top navigation bar includes links for Home, Registrazione, Servizi, Prodotti, Supporto, and Normative. A left sidebar contains options like Benvenuto demo, Cambia Profilo, Modifica Password, Logout, and a Trace Client download link. The main content area is titled 'TABELLA DELLA RINTRACCIABILITÀ LOTTO N° BD02' and features a table of processes and lots. Below this, the 'Lotto selezionato' section provides detailed information for the selected lot, including its code, generation date, product name, and responsible company.

TABELLA DELLA RINTRACCIABILITÀ
LOTTO N° BD02

Processi e relativi lotti lavorati
Lotti da cui ha origine il lotto selezionato tramite i processi di lavorazione sotto riportati. Cliccare su un lotto per risalire ai passaggi precedenti della filiera.

| CODICE LOTTO | DATA GENERAZIONE | NOME PRODOTTO | PROCESSO | FORNITORE | |
|--------------|---------------------|----------------------|----------|-----------------------|--------------------------|
| RC09 | 2006-05-03 11:09:00 | confezioni di cavoli | | Bianchi distribuzioni | Dettagli |

Lotto selezionato
Informazioni sul lotto selezionato.

| | |
|--|--|
| Codice Lotto : | BD02 |
| Data Generazione : | 2006-05-03 11:10:33 |
| Nome Prodotto : | confezioni di cavoli |
| Messaggio di Servizio : | |
| Locazione : | Magazzini di deposito Via Bixio, 44 |
| Azienda Responsabile : | Bianchi distribuzioni |
| Trasformazioni associate al lotto N° RC09 : | |
| Codice Lotto : | RC09 |
| Nome Prodotto : | confezioni di cavoli |
| Trasformazione : | |

Figura 64 Cessione al consumatore

CONCLUSIONI

Il lavoro svolto costituisce una esperienza di modellazione e sviluppo di servizi in contesti di business, e si incentrato su alcuni obiettivi fondamentali:

- Un'attività di modellazione UML dei processi di tracciabilità riguardanti la filiera degli ortaggi, più in particolare del cavolo, a partire da disciplinari di filiera reali
- La validazione di tale modello su di una piattaforma di sistema informativo per la tracciabilità alimentare *Cerere*.
- La modellazione di procedure di tipo amministrativo secondo lo standard ebXML

Grazie alle proprietà di estendibilità della piattaforma di filiera, ed ai pattern del modello *Cerere*, lo schema relativo al contesto degli ortaggi è stato integralmente rappresentato senza necessità di riprogrammare la piattaforma, ma adoperando opportuni documenti XML prodotti a partire dal modello UML progettato nella prima fase.

A livello di usabilità, la piattaforma presenta alcune difficoltà per un utente privo di competenze tecnologiche, nella fase di configurazione, dovute alla necessità di utilizzo del linguaggio XML per la configurazione delle applicazioni e all'inserimento dei codici delle aziende all'interno del database. Tali difficoltà sono in parte sormontabili grazie all'utilizzo di un opportuno editor XML visuale, quale *ebBP Editor*, e di un editor XML sviluppato in precedenti tesi.

Durante le fasi di acquisizione, tracciatura e interrogazione, il sistema risulta invece intuitivo e di facile comprensione per gli operatori di filiera, offrendo anche una agile interfaccia web. Nel complesso, il sistema di tracciabilità, pur avendo richiesto interventi di manutenzione correttiva, operati in linguaggio Java, si è dimostrato soddisfacente da un punto di vista dell'efficacia. Allo stesso modo le informazioni presentate durante le interrogazioni dal portale web, sono risultate coerenti con il modello di partenza.

Per quanto riguarda gli aspetti amministrativi di filiera, la tesi ha consentito di familiarizzare con le problematiche relative allo svolgimento di transazioni elettroniche e soprattutto di approfondire la conoscenza delle business collaboration di ebXML, evidenziando nuove potenzialità per la piattaforma *Cerere*.

Lo standard, pur nascendo come strumento di modellazione dei processi commerciali, ben si estende ai processi produttivi ed organizzativi di filiera, offrendo diversi livelli di affidabilità e sicurezza e vantando anche il pregio di non richiedere investimenti ingenti per le piccole

imprese, che di solito caratterizzano il primo segmento della filiera alimentare, grazie ad una ampia organizzazione non-profit che offre applicativi software free.

Infine, è stato integrato al sistema *Cerere* un applicativo che consente lo scambio di messaggi di business tra gli operatori, *ebMail*, offrendo una semplice interfaccia utente, simile ad un client di posta elettronica, e semplicità di configurazione ed estendibilità grazie a plug-in sviluppati in linguaggio Java.

A livello tecnologico, i fattori critici che hanno consentito alla piattaforma di operare su contesti differenti sono la modularità (ciascun modulo è finalizzato al supporto di specifiche categorie di processi), estendibilità (si possono aggiungere nuove funzionalità) e l'altissimo livello di integrazione (può connettersi a diverse piattaforme per il trasferimento di dati e servizi) e di interoperabilità (può co-operare con diverse piattaforme adeguandosi a differenti semantiche di business).

A livello metodologico, l'estendibilità della piattaforma su nuovi contesti di business è garantita se vengono rispettati i seguenti fattori critici: i) la possibilità di eseguire delle simulazioni su un supporto integrale al paradigma ebXML; ii) la formulazione di un modello UML-XML in grado di specificare tutti gli aspetti di "run-time", a partire dai disciplinari produttivi e dai regolamenti di controllo di filiera.

Tutto ciò consente di separare nettamente le competenze di produzione, da quelle gestionali ed informatiche nei processi di modellazione, permettendo lo sviluppo di sistemi informatici facilmente "plasmabili" ed estendibili a partire dai processi aziendali.

Una tale architettura risulta fondamentale in servizi di tracciabilità, di natura trasversale ai singoli sistemi informativi aziendali, in contesti caratterizzati da forte eterogeneità.

Appendice A

Script di configurazione SQL

use trace;

```
INSERT INTO ATTORE_RESPONSABILE (RAGIONE_SOCIALE , RECAPITO , CAP,
CITTA, TELEFONO, FAX , EMAIL, PREFISSO_AZIENDA)
VALUES
('Vivai Landi','Via Ginori,
80','57023','Cecina','0586660001','05000002','info@landi.it','VL00001');
```

```
INSERT INTO ATTORE_RESPONSABILE (RAGIONE_SOCIALE , RECAPITO , CAP,
CITTA, TELEFONO, FAX , EMAIL, PREFISSO_AZIENDA)
VALUES
('Azienda agricola Maestrale','Via
metauro,12','57013','Rosignano','05093523','05867876','info@maestrale.it','AM00002');
```

```
INSERT INTO ATTORE_RESPONSABILE (RAGIONE_SOCIALE , RECAPITO , CAP,
CITTA, TELEFONO, FAX , EMAIL, PREFISSO_AZIENDA)
VALUES
('Rossini confezionamenti','Via Roma,
11','56100','PISA','05013458','05012341','info@rossi.it','RC00002');
```

```
INSERT INTO ATTORE_RESPONSABILE (RAGIONE_SOCIALE , RECAPITO , CAP,
CITTA, TELEFONO, FAX , EMAIL, PREFISSO_AZIENDA)
VALUES
('Bianchi distribuzioni','Piazza Grande,
115','56100','PISA','05033321','05023334','info@bianchidistribuzioni.it','BD00001');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('12','Campo aperto','Via Mascagni, 88','57023','Cecina','MVL01');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('12','Serra A','Via Mascagni, 88','57023','Cecina','MVL02');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('12','Serra B','Via Mascagni, 90','57023','Cecina','MVL03');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('12','Magazzino sementi','Via Mascagni, 90','57023','Cecina','MVL04');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, CODICE_SITO)
VALUES ('12','Furgone','MVL05');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('13','Terreno A','Via Metauro, 12','57013','Rosignano Marittimo','MAM01');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('13','Terreno B','Via Tronto, 23','57023','Cecina','MAM02');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('13','Magazzino di raccolta','Via Metauro, 12','57013','Rosignano
Marittimo','MAM03');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, CODICE_SITO)
VALUES ('14','Furgone','MRC01');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('14','Linea di lavorazione','Via Amendola, 27','57100','Livorno','MRC02');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('14','Cella frigorifera','Via Amendola, 27','57100','Livorno','MRC03');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, INDIRIZZO, CAP, CITTA,
CODICE_SITO)
VALUES ('15','Magazzini di deposito','Via Bixio, 44','56100','PISA','MBD01');
```

```
INSERT INTO SITO (ATTORE_RESP_ID, DESCRIZIONE, CODICE_SITO)
VALUES ('15','Furgone coibentato','Via Tanaro, 102','56100','PISA','10001');
```

```
INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('Kg','Peso');
```

```
INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('C','Numero piantine presenti nel contenitore');
```

```
INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('Kg/ha','Azoto totale immesso in terreno');
```

```
INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('Kg/ha','Fosforo totale immesso in terreno');
```

```
INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('Kg/ha','Potassio totale immesso in terreno');
```


INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('Kg/ha','Magnesio totale immesso in terreno');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('','Danneggiamento medio delle palle');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('','Consistenza media delle palle');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('','Numero confezioni');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('','Categoria');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('°C','Temperatura di conservazione');

INSERT INTO DESCRIZIONE_MISURAZIONE (UNITA , SEMANTICA)
VALUES ('°C','Temperatura di trasporto')

Appendice B

Files di configurazione dei Client

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Document      : config.xml
    Created on    : 27 Febbraio 2006, 20.37
    Author       : Marco
    Description: file di configurazione per traceClient Landi vivai
-->
<config code="VL00001" name="Vivai Landi">
    <ActorList>
        <Actor code="0000000" name="NATURA" type="supplier"/>
        <Actor code="AM00002" name="Azienda agricola Maestrale"
type="customer"/>
    </ActorList>
    <SiteList>
        <Site code="MVL02" name="Serra A"/>
        <Site code="MVL04" name="Magazzino sementi"/>
        <Site code="MVL01" name="Terreno"/>
        <Site code="MVL03" name="Serra B"/>
        <Site code="MVL05" name="Furgone"/>
    </SiteList>
    <MeasureTypeList>
        <Measure unit="Kg" semantic="Peso"/>
        <Measure unit="" semantic="Numero di piantine presenti nel
contenitore"/>
    </MeasureTypeList>
</config>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Document      : config.xml
    Created on    : 27 Febbraio 2006, 20.37
    Author       : Marco
    Description: file di configurazione per traceClient Landi vivai
-->
<config code="AM00002" name="Azienda agricola Maestrale">
    <ActorList>
        <Actor code="VL00001" name="Vivai Landi" type="supplier"/>
        <Actor code="RC00002" name="Rossini confezionamenti"
type="customer"/>
    </ActorList>
    <SiteList>
        <Site code="MAM01" name="Terreno A"/>
        <Site code="MAM03" name="Magazzino di raccolta"/>
        <Site code="MAM02" name="Terreno B"/>
    </SiteList>
    <MeasureTypeList>
        <Measure unit="Kg/ha" semantic="Azoto totale immesso in
terreno"/>
    </MeasureTypeList>
</config>
```

```

        <Measure unit="Kg/ha" semantic="Magnesio totale immesso in
terreno"/>
        <Measure unit="Kg/ha" semantic="Potassio totale immesso in
terreno"/>
        <Measure unit="Kg/ha" semantic="Fosforo totale immesso in
terreno"/>
        <Measure unit="" semantic="Numero di piantine presenti nel
contenitore"/>
        <Measure unit="" semantic="Consistenza media delle palle"/>
        <Measure unit="" semantic="Danneggiamento medio delle palle"/>
        <Measure unit="Kg" semantic="Peso"/>
    </MeasureTypeList>
</config>

```

```

<?xml version="1.0" encoding="utf-8"?>
<!--
    Document      : config.xml
    Created on    : 27 Febbraio 2006, 20.37
    Author       : Marco
    Description: file di configurazione per traceClient Landi vivai
-->
<config code="RC00002" name="Rossini confezionamenti">
    <ActorList>
        <Actor code="0000000" name="NATURA" type="customer"/>
        <Actor code="AM00002" name="Azienda agricola Maestrale"
type="supplier"/>
        <Actor code="BD00001" name="Bianchi distribuzioni"
type="customer"/>
    </ActorList>
    <SiteList>
        <Site code="MRC03" name="Cella frigorifera"/>
        <Site code="MRC01" name="Furgone"/>
        <Site code="MRC02" name="Linea di lavorazione"/>
    </SiteList>
    <MeasureTypeList>
        <Measure unit="" semantic="Numero confezioni"/>
        <Measure unit="Kg" semantic="Peso"/>

        <Measure unit="" semantic="Categoria"/>
    </MeasureTypeList>
</config>

```

```

<?xml version="1.0" encoding="utf-8"?>
<!--
    Document      : config.xml
    Created on    : 27 Febbraio 2006, 20.37
    Author       : Marco
    Description: file di configurazione per traceClient Landi vivai
-->
<config code="BD00001" name="Bianchi distribuzioni">
    <ActorList>
        <Actor code="C000000" name="Consumatore finale"
type="customer"/>
        <Actor code="RC00001" name="Rossini confezionamenti"
type="supplier"/>
    </ActorList>
    <SiteList>
        <Site code="MBD01" name="Magazzini di deposito"/>
        <Site code="MBD02" name="Furgone coibentato"/>
    </SiteList>
</config>

```

```
</SiteList>
<MeasureTypeList>
  <Measure unit="" semantic="Numero confezioni"/>
  <Measure unit="Kg" semantic="Peso"/>
  <Measure unit="C" semantic="Temperatura di conservazione"/>
  <Measure unit="C" semantic="Temperatura di trasporto"/>
</MeasureTypeList>
</config>
```

Appendice C

Schema XML dei messaggi

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ComponentType">
    <xs:all minOccurs="0">
      <xs:element ref="ServiceMsg" minOccurs="0"/>
      <xs:element ref="Description" minOccurs="0"/>
    </xs:all>
    <xs:sequence>
      <xs:element name="Measure" type="MeasureType"
minOccurs="0"
                maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="location" type="xs:string" use="required"/>
    <xs:attribute name="date" type="xs:string" use="required"/>
    <xs:attribute name="value" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:element name="Description" type="xs:string"/>
  <xs:complexType name="FromType">
    <xs:attribute name="location" type="xs:string"/>
    <xs:attribute name="actor" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="LotType">
    <xs:sequence>
      <xs:element name="Component" type="ComponentType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:all minOccurs="0">
      <xs:element name="ProcessingProperty"
type="ProcessingPropertyType" minOccurs="0"/>
      <xs:element name="LotProperty" type="LotPropertyType"
minOccurs="0"/>
    </xs:all>
    <xs:attribute name="value" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="LotPropertyType">
    <xs:all minOccurs="0">
      <xs:element ref="ServiceMsg" minOccurs="0"/>
      <xs:element ref="Description" minOccurs="0"/>
    </xs:all>
    <xs:sequence>
      <xs:element name="Measure" type="MeasureType"
minOccurs="0"
                maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MeasureType">
    <xs:all>
```

```

        <xs:element ref="Value"/>
        <xs:element ref="Semantic"/>
        <xs:element ref="Type"/>
    </xs:all>
</xs:complexType>

<xs:complexType name="ProcessingPropertyType">
    <xs:all minOccurs="0">
        <xs:element ref="ServiceMsg" minOccurs="0"/>
        <xs:element ref="Description" minOccurs="0"/>
    </xs:all>
    <xs:sequence>
        <xs:element name="Measure" type="MeasureType"
minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Semantic" type="xs:string"/>
<xs:element name="ServiceMsg" type="xs:string"/>
<xs:complexType name="ToType">
    <xs:attribute name="location" type="xs:string"/>
    <xs:attribute name="actor" type="xs:string" use="required"/>
</xs:complexType>
<xs:element name="TraceLot">
    <xs:complexType>
        <xs:all>
            <xs:element name="Lot" type="LotType"/>
            <xs:element name="To" type="ToType"/>
            <xs:element name="From" type="FromType"/>
        </xs:all>
        <xs:attribute name="date" type="xs:string"
use="required"/>
        <xs:attribute name="state" type="xs:string"
            use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="Type" type="xs:string"/>
<xs:element name="Value" type="xs:string"/>
</xs:schema>

```

Appendice D

Schema XML del CPA

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Nell'elemento di root è necessario definire i namespace a l'id del CPA -->
<tp:CollaborationProtocolAgreement
  xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-
2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-
cpa-2_0.xsd
                      cpp-cpa-2_0a.xsd"
  tp:cpaid="CPA_2006"
  tp:version="2.0a">

  <!-- Lo status può assumere tre valori
    - proposed = il corrente CPA è una proposta ma deve ancora essere concordato
    - agreed = il CPA è stato accordato ma non ancora firmato
    - signed = il CPA è stato firmato in data <Start> ed è valido fino alla data
  <End>
  -->
  <tp:Status tp:value="signed"/>
  <tp:Start>2004-04-04T18:39:09</tp:Start>
  <tp:End>2006-06-06T18:40:00</tp:End>

  <!-- ConversationConstraints può apparire al più una volta.
    Permette di definire il numero massimo di conversazioni dopo delle quali è
    necessaria una rinegoziazione del CPA (invocationLimit) e il numero massimo
    di conversazioni contemporanee (concurrentConversations)
  -->
  <tp:ConversationConstraints
    tp:invocationLimit="100"
    tp:concurrentConversations="4"/>

  <!-- Nel CPA vanno definiti due campi PartyInfo, uno per ogniuna delle due parti
    che hanno stipulato questo accordo.
    E' possibile definire un nome e gli id del Channel e del Package di default
    definiti
    più avanti nel codice.
  -->
  <tp:PartyInfo
    tp:partyName="VivaiLandi"
    tp:defaultMshChannelId="TraceChannel"
    tp:defaultMshPackageId="TracePackaging">

  <!-- E' necessario definire l'id dell'azienda che può essere in vari formati.
  -->
    <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:ean_ucc">
      VL00001
    </tp:PartyId>

  <!-- E' possibile definire un link ad un documento esterno che fornisce
    informazioni
    più dettagliate riguardo all'azienda.
  -->
```

```

    <tp:PartyRef
      xlink:type="simple"
      xlink:href="http://www.vivailandi.com/info.xml"/>

<!-- Bisogna specificare il ruolo assunto dall'azienda in questione all'interno
della
      collaborazione e i documenti che è autorizzato ad inviare ed a ricevere.
-->
    <tp:CollaborationRole>
<!-- In ProcessSpecification va inserito il collegamento ad un file XML che
definisce
      le specifiche del Business Process. Nel Nostro esempio non è utilizzato e per
il
      suo uso si rimanda alla documentazione fornita da OASIS e da UNCEFACT
(www.ebxml.org)
-->
      <tp:ProcessSpecification
        tp:version="2.0a"
        tp:name="SendTraceability"
        xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/registry/thin/ortaggi/TraceBPS.x
ml"/>

<!-- Nel nostro caso il ruolo definito prende il nome di TraceActor.
      I dettagli di questo ruolo vanno specificati nel Business Process ma non è
necessario
      nel nostro prototipo.
-->
      <tp:Role
        tp:name="TraceActor"
        xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/registry/thin/ortaggi/TraceBPS.x
ml#TraceActorId"/>

<!-- Il ServiceBinding definisce il traffico di messaggi che possono essere inviati
o ricevuti
      attraverso il DeliveryChannel scelto, la cui definizione è più avanti nel
codice.
-->
      <tp:ServiceBinding>
<!-- Il campo Service è molto importante nel nostro prototipo, infatti esso
fornisce le
      informazioni di routing per l'ebXML Message Header. Serve per indirizzare i
messaggi
      ricevuti al corretto entry point dell'applicazione.
      Il Service viene definito in dettaglio nel BP (dovrebbe corrispondere al uuid
dell'elemento processSpecification.
-->
      <tp:Service tp:type="anyURI">urn:TraceService</tp:Service>

<!-- E' possibile definire i documenti che è possibile inviare. Nel nostro caso,
      l'attore responsabile può inviare un unico documento che contiene le
necessarie
      informazioni per la tracciabilità della filiera.
      TraceDocumentPackage è l'id del campo Packaging definito più sotto che
definisce
      alcune informazioni sui contenuti del messaggio SOAP.
      E' possibile avere più di un campo CanSend.
-->
      <tp:CanSend>
        <tp:ThisPartyActionBinding
          tp:id="VivaiLandi_STD"
          tp:action="Send Trace Document Action"
          tp:packageId="TracePackaging">

<!-- E' possibile definire le necessarie informazioni per l'implementazione dei
protocolli di sicurezza. Per informazioni più dettagliate su questo campo si
rimanda alla documentazione fornita nel sito www.ebxml.org.

```


Nel nostro prototipo le informazioni sulla sicurezza sono inserite manualmente nel codice sorgente e non vengono prelevate da questo CPA, pertanto questo campo non viene da noi utilizzato.

```
-->
    <tp:BusinessTransactionCharacteristics
      tp:isNonRepudiationRequired="true"
      tp:isNonRepudiationReceiptRequired="true"
      tp:isConfidential="transient"
      tp:isAuthenticated="persistent"
      tp:isTamperProof="persistent"
      tp:isAuthorizationRequired="true"
      tp:timeToAcknowledgeReceipt="PT2H"
      tp:timeToPerform="P1D"/>

<!-- Queste informazioni collegano questo documento con la o le collaborazioni
binarie
      definite nel documento di specifica del BP.
-->
    <tp:ActionContext
      tp:binaryCollaboration="Send Trace Document"
      tp:businessTransactionActivity="Send"
      tp:requestOrResponseAction="Send Trace Document Action"/>

<!-- ChannelId contiene l'id del canale di trasmissione utilizzato per l'invio del
documento. Questo campo è definito più in basso.
-->
    <tp:ChannelId>TraceChannel</tp:ChannelId>
  </tp:ThisPartyActionBinding>
</tp:CanSend>
</tp:ServiceBinding>
</tp:CollaborationRole>

<!-- In questo campo è necessario definire le informazioni sul certificato che
andranno poi
      inserite nei messaggi SOAP.
-->
    <tp:Certificate tp:certId="VivaiLandiSigningCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_SigningCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="TrustedRootCertA1">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA1_Key </ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="TrustedRootCertA2">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA2_Key </ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="TrustedRootCertA4">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA4_Key </ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>
    <tp:SecurityDetails tp:securityId="CompanyA_TransportSecurity">
      <tp:TrustAnchors>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA1"/>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA2"/>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA4"/>
      </tp:TrustAnchors>
    </tp:SecurityDetails>

<!-- Il campo DeliveryChannel collega un elemento Transport ad un elemento
DocExchange
      e definisce le caratteristiche di comunicazione
-->
    <tp:DeliveryChannel
```

```

        tp:channelId="TraceChannel"
        tp:transportId="TraceTransport"
        tp:docExchangeId="TraceDocExchange">

<!-- Questo campo descrive alcune informazioni riguardanti i messaggi di
Acknowledgement.
Anche per questo ri rimanda alla documentazione di ebxml.
Nel nostro prototipo non vengono trattate.
-->
        <tp:MessagingCharacteristics
            tp:syncReplyMode="mshSignalsOnly"
            tp:ackRequested="always"
            tp:ackSignatureRequested="always"
            tp:duplicateElimination="always"
            tp:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"/>
        </tp:DeliveryChannel>

<!-- Il campo Transport definisce i protocolli di trasporto e di sicurezza
utilizzati. -->
        <tp:Transport tp:transportId="TraceTransport">

<!-- Il campo TransportSender, se presente, definisce la capacità di inviare
messaggi
attraverso il protocollo specificato.
-->
        <tp:TransportSender>
            <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
            <tp:TransportClientSecurity>
                <tp:TransportSecurityProtocol tp:version="3.0">
                    SSL
                </tp:TransportSecurityProtocol>
                <tp:ClientCertificateRef tp:certId="VivaiLandiSigningCert"/>
            </tp:TransportClientSecurity>
        </tp:TransportSender>
    </tp:Transport>

<!-- Il campo DocExchange fornisce informazioni utili per lo scambio dei messaggi e
le
proprietà del Message Service.
-->
        <tp:DocExchange tp:docExchangeId="TraceDocExchange">
            <tp:ebXMLSenderBinding tp:version="2.0">

<!-- In questo campo è possibile definire il numero di tentativi per l'invio di un
messaggio e l'intervallo tra un tentativo e l'altro (PT2H = Periodo di 2 ore)
-->
            <tp:ReliableMessaging>
                <tp:Retries>5</tp:Retries>
                <tp:RetryInterval>PT2H</tp:RetryInterval>
                <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
            </tp:ReliableMessaging>

<!-- Qui è possibile inserire informazioni sui protocolli per la firma digitale. --
>
            <tp:SenderNonRepudiation>
                <tp:NonRepudiationProtocol>
                    http://www.w3.org/2000/09/xmldsig#
                </tp:NonRepudiationProtocol>
                <tp:HashFunction>
                    http://www.w3.org/2000/09/xmldsig#sha1
                </tp:HashFunction>
                <tp:SignatureAlgorithm>
                    http://www.w3.org/2000/09/xmldsig#dsa-sha1
                </tp:SignatureAlgorithm>
                <tp:SigningCertificateRef tp:certId="VivaiLandiSigningCert"/>
            </tp:SenderNonRepudiation>

<!-- Qui è possibile inserire informazioni sui protocolli per il crittaggio dei
messaggi. -->
            <tp:SenderDigitalEnvelope>

```

```

        <tp:DigitalEnvelopeProtocol tp:version="2.0">
            S/MIME
        </tp:DigitalEnvelopeProtocol>
        <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>
        <tp:EncryptionSecurityDetailsRef
            tp:securityId="VivaiLandiSigningCert"/>
        </tp:SenderDigitalEnvelope>
    </tp:ebXMLSenderBinding>
</tp:DocExchange>
</tp:PartyInfo>

<tp:PartyInfo
    tp:partyName="Coordinator"
    tp:defaultMshChannelId="TraceChannel2"
    tp:defaultMshPackageId="TracePackaging">
    <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:ean_ucc">
        3047264
    </tp:PartyId>
    <tp:PartyRef
        xlink:type="simple"
        xlink:href="http://www.coordinatore.com/info.xml"/>
    <tp:CollaborationRole>
        <tp:ProcessSpecification
            tp:version="2.0a"
            tp:name="ReceiveTraceability"
            xlink:type="simple"
            xlink:href="http://www.coordinatore.com/processes/TraceBPS.xml"/>
        <tp:Role
            tp:name="Coordinator"
            xlink:type="simple"

xlink:href="http://www.coordinatore.com/processes/TraceBPS.xml#CoordinatorId"/>
        <tp:ServiceBinding>
            <tp:Service tp:type="anyURI">urn:TraceService</tp:Service>

<!-- Questo campo è simile al CanSend utilizzato nel PartyId dell'attore
responsabile
    ma serve per definire i messaggi che è possibile ricevere.
-->
        <tp:CanReceive>
            <tp:ThisPartyActionBinding
                tp:id="Coordinatr_RTD"
                tp:action="Receive Trace Document Action"
                tp:packageId="TracePackaging">
                <tp:BusinessTransactionCharacteristics
                    tp:isNonRepudiationRequired="true"
                    tp:isNonRepudiationReceiptRequired="true"
                    tp:isConfidential="transient"
                    tp:isAuthenticated="persistent"
                    tp:isTamperProof="persistent"
                    tp:isAuthorizationRequired="true"
                    tp:timeToAcknowledgeReceipt="PT2H"
                    tp:timeToPerform="P1D"/>
                <tp:ActionContext
                    tp:binaryCollaboration="Receive Trace Document"
                    tp:businessTransactionActivity="Receive Trace Document"
                    tp:requestOrResponseAction="Receive Trace Document
Action"/>
            </tp:ThisPartyActionBinding>
        </tp:CanReceive>
    </tp:ServiceBinding>
</tp:CollaborationRole>
<tp:Certificate tp:certId="CoordinatorSigningCert">
    <ds:KeyInfo>
        <ds:KeyName>CompanyB_SigningCert_Key</ds:KeyName>
    </ds:KeyInfo>
</tp:Certificate>
<tp:SecurityDetails tp:securityId="CompanyB_TransportSecurity">
    <tp:TrustAnchors>

```

```

        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA1"/>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA2"/>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA4"/>
    </tp:TrustAnchors>
</tp:SecurityDetails>
<tp:DeliveryChannel
    tp:channelId="TraceChannel2"
    tp:transportId="TraceTransport2"
    tp:docExchangeId="TraceDocExchange2">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="mshSignalsOnly"
        tp:ackRequested="always"
        tp:ackSignatureRequested="always"
        tp:duplicateElimination="always"
        tp:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"/>
    </tp:DeliveryChannel>
<tp:Transport tp:transportId="TraceTransport2">
    <tp:TransportReceiver>
        <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
        <tp:Endpoint
            tp:uri="http://cerere04.iet.unipi.it:8080/msh/"
            tp:type="allPurpose"/>
        <tp:TransportServerSecurity>
            <tp:TransportSecurityProtocol tp:version="3.0">
                SSL
            </tp:TransportSecurityProtocol>
            <tp:ServerCertificateRef tp:certId="CompanyB_TransportSecurity"/>
        </tp:TransportServerSecurity>
    </tp:TransportReceiver>
</tp:Transport>
<tp:DocExchange tp:docExchangeId="TraceDocExchange2">
    <tp:ebXMLReceiverBinding tp:version="2.0">
        <tp:ReliableMessaging>
            <tp:Retries>5</tp:Retries>
            <tp:RetryInterval>PT2H</tp:RetryInterval>
            <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
        </tp:ReliableMessaging>
        <tp:ReceiverNonRepudiation>
            <tp:NonRepudiationProtocol>
                http://www.w3.org/2000/09/xmldsig#
            </tp:NonRepudiationProtocol>
            <tp:HashFunction>
                http://www.w3.org/2000/09/xmldsig#sha1
            </tp:HashFunction>
            <tp:SignatureAlgorithm>
                http://www.w3.org/2000/09/xmldsig#dsa-sha1
            </tp:SignatureAlgorithm>
            <tp:SigningSecurityDetailsRef
tp:securityId="CompanyB_TransportSecurity"/>
            </tp:ReceiverNonRepudiation>
            <tp:ReceiverDigitalEnvelope>
                <tp:DigitalEnvelopeProtocol tp:version="2.0">
                    S/MIME
                </tp:DigitalEnvelopeProtocol>
                <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>
                <tp:EncryptionCertificateRef
                    tp:certId="CoordinatorSigningCert"/>
            </tp:ReceiverDigitalEnvelope>
        </tp:ebXMLReceiverBinding>
    </tp:DocExchange>
</tp:PartyInfo>

<!-- E' possibile definire i Content Type che costituiscono i messaggi -->
<tp:SimplePart tp:id="SP1" tp:mimetype="text/xml"/>
<tp:SimplePart tp:id="SP2" tp:mimetype="*/xml"/>

<!-- Il campo Packaging evidenzia le varie componenti del messaggio SOAP da inviare -->
<tp:Packaging tp:id="TracePackaging">
    <tp:ProcessingCapabilities tp:generate="true" tp:parse="true"/>

```

```

<!-- Il campo CompositeList indica come combinare i SimplePart -->
  <tp:CompositeList>

<!-- composite contiene un campo crittato che, nel nostro caso, è il messaggio
di tracciabilità che è di tipo SP1 ed un altro messaggio che viene utilizzato
per
le informazioni sulla qualità sempre di tipo SP1 (text/xml)
-->
    <tp:Composite
      tp:id="EP1"
      tp:mimetype="text/xml"
      tp:mimeparameters="smime-type='enveloped-data'">
      <tp:Constituent tp:idref="SP1"/>
      <tp:Constituent tp:idref="SP1"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
</tp:CollaborationProtocolAgreement>

```

Appendice E

Schema XML del Business Process

```
<ProcessSpecification name="Simple" version="1.1">
  <BusinessDocument name="SendTraceability"/>
  <Package name="TracePackaging">
    <BinaryCollaboration name="Send Trace Document">
      <InitiatingRole name="TraceActor"/>
      <RespondingRole name="Coordinator"/>
      <BusinessTransactionActivity name="Send"
        businessTransaction="Send"
        fromAuthorizedRole="TraceActor"
        toAuthorizedRole="Coordinator"
        isConcurrent="true"/>
    </BinaryCollaboration>

    <BusinessTransaction name="Send"
      isGuaranteedDeliveryRequired="true">
      <RequestingBusinessActivity name=""
        isAuthorizationRequired="true"
        isNonRepudiationReceiptRequired="true"
        isNonRepudiationRequired="true">
        <DocumentEnvelope isPositiveResponse="true"
          isConfidential="true"
          isTamperProof="true"
          isAuthenticated="true"
          businessDocument="SendTraceability"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name=""
          isIntelligibleCheckRequired="true"
          isNonRepudiationReceiptRequired="true"
          isNonRepudiationRequired="true">
          </RespondingBusinessActivity>
        </BusinessTransaction>
      </Package>
    </ProcessSpecification>
```

Appendice F

Business Process per l'ingresso in filiera

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification >
<ProcessSpecification
  name="ingresso filiera"
  version="1.2"
  uuid="urn:TraceService">
<Documentation>questo business process regola l'ingresso in filiera
da parte di un attore responsabile,
lo scambio di documenti avviene tra l'attore responsabile stesso e il
coordinatore della filiera</Documentation>
  <BusinessDocument
    name="info richiesta"
    nameID="ir1"/>

  <BusinessDocument
    name="esito verifica"
    nameID="ev1" />

  <BusinessDocument
    name="data verifica"
    nameID="dv1" />

  <BusinessDocument
    name="modifiche effettuate"
    nameID="me1" />

  <Package
    name="ingresso filiera">
    <BinaryCollaboration
      name="richiesta ingresso"
      timeToPerform="P3D">
      <InitiatingRole
        name="richiedente" />

      <RespondingRole
        name="coordinatore" />

      <BusinessTransactionActivity
        name="richiesta ingresso"
        businessTransaction="richiesta ingresso"
        fromAuthorizedRole="richiedente"
        toAuthorizedRole="coordinatore" />

      <BusinessTransactionActivity
        name="invio esito"
        businessTransaction="invio esito"
        fromAuthorizedRole="coordinatore"
        toAuthorizedRole="richiedente" />

    <Start toBusinessState="richiesta ingresso"/>
  </BinaryCollaboration>
</Package>
</ProcessSpecification>
</Transition>
```

```
formBusinessState="richiesta ingresso"
toBusinessState="invio esito"/>

<Success formBusinessState="invio esito"
conditionGuard="Success"/>

</BinaryCollaboration>

<BusinessTransaction
name="richiesta ingresso">
  <RequestingBusinessActivity
name=""
isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P2D">
    <DocumentEnvelope
isPositiveResponse="true"
businessDocument="info richiesta" />
  </RequestingBusinessActivity>

  <RespondingBusinessActivity
name=""
isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P1D">
    <DocumentEnvelope
isPositiveResponse="true"
businessDocument="data verifica" />
  </RespondingBusinessActivity>
</BusinessTransaction>

<BusinessTransaction
name="invio esito">
  <RequestingBusinessActivity
name=""
isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P1D">
    <DocumentEnvelope
isPositiveResponse="true"
businessDocument="esito verifica" />
  </RequestingBusinessActivity>

  <RespondingBusinessActivity
name=""
isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P1D">
    <DocumentEnvelope
isPositiveResponse="true"
businessDocument="modifiche effettuate" />
  </RespondingBusinessActivity>
</BusinessTransaction>
</Package>
</ProcessSpecification>
```


Appendice G

Business Process per l'ordine delle piantine

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification SYSTEM "ebXMLProcessSpecification-
v1.00.dtd">
<ProcessSpecification
  name="acquisto piante"
  version="1.2"
  uuid="[1234-5678]">
  <BusinessDocument
    name="ordine" />

  <BusinessDocument
    name="info recapito" />

  <BusinessDocument
    name="ordine accettato" />

  <BusinessDocument
    name="ordine rifiutato" />

  <BusinessDocument
    name="info pagamento" />

  <Package
    name="acquisto piante">
    <BinaryCollaboration
      name="acquisto piante">
      <InitiatingRole
        name="acquirente" />

      <RespondingRole
        name="venditore" />

      <BusinessTransactionActivity
        name="ordine acquisto"
        businessTransaction="ordine acquisto"
        fromAuthorizedRole="acquirente"
        toAuthorizedRole="venditore" />

      <BusinessTransactionActivity
        name="invio pagamento"
        businessTransaction="Invio pagamento"
        fromAuthorizedRole="acquirente"
        toAuthorizedRole="venditore" />

      <Start toBusinessState="ordine acquisto"/>
      <Transition
        formBusinessState="ordine acquisto"
        •toBusinessState="invio pagamento"/>

      <Success formBusinessState="invio pagamento"
        conditionGuard="Success"/>
```

```
</BinaryCollaboration>

<BusinessTransaction
  name="ordine acquisto">
  <RequestingBusinessActivity
    name=""
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="P1D"
    timeToAcknowledgeAcceptance="P2D">
    <DocumentEnvelope
      isPositiveResponse="true"
      businessDocument="ordine">
      <Attachment
        name="note di consegna"
        businessDocument="info recapito" />
      </DocumentEnvelope>
    </RequestingBusinessActivity>

    <RespondingBusinessActivity
      name=""
      isNonRepudiationRequired="true"
      timeToAcknowledgeReceipt="P1D">
      <DocumentEnvelope
        isPositiveResponse="true"
        businessDocument="esito ordine" />
      </RespondingBusinessActivity>
    </BusinessTransaction>

  <BusinessTransaction
    name="invio pagamento">
    <RequestingBusinessActivity
      name=""
      isNonRepudiationRequired="true"
      timeToAcknowledgeReceipt="P1D">
      <DocumentEnvelope
        isPositiveResponse="true"
        businessDocument="info pagamento" />
      </RequestingBusinessActivity>

    <RespondingBusinessActivity
      name=""
      isNonRepudiationRequired="true"
      timeToAcknowledgeReceipt="P1D">
      <DocumentEnvelope
        isPositiveResponse="true"
        businessDocument="ack" />
      </RespondingBusinessActivity>
    </BusinessTransaction>
  </Package>
</ProcessSpecification>
```

Appendice H

CPA per l'ingresso in filiera

```

<?xml version="1.0"?>
<!-- Copyright UN/CEFACT and OASIS, 2001. All Rights Reserved. -->
<tp:CollaborationProtocolAgreement
  xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-
2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd cpp-cpa-2_0a.xsd"
  tp:cpaid="uri:companyA-and-companyB-cpa"
  tp:version="2_0a">
  <tp:Status
    tp:value="signed" />

  <tp:Start>2005-05-20T07:21:00Z</tp:Start>

  <tp:End>2007-05-20T07:21:00Z</tp:End>

  <tp:ConversationConstraints
    tp:invocationLimit="100"
    tp:concurrentConversations="10" />

<!-- Party info for CompanyA -->
<tp:PartyInfo
  tp:partyName="Vivai Landi"
  tp:defaultMshChannelId="asyncChannelA1"
  tp:defaultMshPackageId="CompanyA_MshSignalPackage">
  <tp:PartyId>VL00001</tp:PartyId>

  <tp:PartyRef
    xlink:href="http://www.landivivai.it/info.html" />

  <tp:CollaborationRole>
    <tp:ProcessSpecification
      tp:version="1.2"
      tp:name="ingresso filiera"
      xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s1.xml"
      tp:uuid="urn:TraceService" />

    <tp:Role
      tp:name="richiedente"
      xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s1.xml#richiedente" />

    <tp:ApplicationCertificateRef
      tp:certId="CompanyA_AppCert" />

```

```

    <tp:ServiceBinding>
      <tp:Service>bpid:icann:iet.unipi.it:3A4$2.0</tp:Service>

      <tp:CanSend>
        <tp:ThisPartyActionBinding
          tp:id="companyA_ABID1"
          tp:action="info_richiesta"
          tp:packageId="CompanyA_RequestPackage">
          <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true"
            tp:timeToAcknowledgeReceipt="PT2H" />

          <tp:ActionContext
            tp:binaryCollaboration="richiesta ingresso"
            tp:businessTransactionActivity="richiesta ingresso"
            tp:requestOrResponseAction="a" />

          <tp:ChannelId>asyncChannelA1</tp:ChannelId>
        </tp:ThisPartyActionBinding>

    <tp:OtherPartyActionBinding>companyB_ABID4</tp:OtherPartyActionBinding>
  </tp:CanSend>

  <tp:CanSend>
    <tp:ThisPartyActionBinding
      tp:id="companyA_ABID2"
      tp:action="ack"
      tp:packageId="CompanyA_ReceiptAcknowledgmentPackage">
      <tp:BusinessTransactionCharacteristics
        tp:isNonRepudiationRequired="true"
        tp:isConfidential="transient"
        tp:isAuthenticated="persistent"
        tp:isTamperProof="none"
        tp:isAuthorizationRequired="true" />

      <tp:ChannelId>asyncChannelA1</tp:ChannelId>

      <tp:ActionContext
        tp:binaryCollaboration="richiesta ingresso"
        tp:businessTransactionActivity="richiesta ingresso"
        tp:requestOrResponseAction="aaa" />
    </tp:ThisPartyActionBinding>

  <tp:OtherPartyActionBinding>companyB_ABID5</tp:OtherPartyActionBinding>
</tp:CanSend>

  <tp:CanSend>
    <tp:ThisPartyActionBinding
      tp:id="companyA_ABID3"
      tp:action="modifiche_effettuate"
      tp:packageId="CompanyA_modPackage">
      <tp:BusinessTransactionCharacteristics
        tp:isNonRepudiationRequired="true"
        tp:isConfidential="transient"
        tp:isAuthenticated="persistent"
        tp:isTamperProof="persistent"
        tp:isAuthorizationRequired="true"

```

```

        tp:timeToAcknowledgeReceipt="PT4H" />

        <tp:ActionContext
            tp:binaryCollaboration="richiesta ingresso"
            tp:businessTransactionActivity="invio esito"
            tp:requestOrResponseAction="aaa" />

        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID6</tp:OtherPartyActionBinding>
</tp:CanSend>

<tp:CanReceive>
    <tp:ThisPartyActionBinding
        tp:id="companyA_ABID4"
        tp:action="ack"
        tp:packageId="CompanyA_ReceiptAcknowledgmentPackage">
        <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="persistent"
            tp:isAuthorizationRequired="true"
            tp:timeToAcknowledgeReceipt="PT2H" />

        <tp:ActionContext
            tp:binaryCollaboration="richiesta ingresso"
            tp:businessTransactionActivity="richiesta ingresso"
            tp:requestOrResponseAction="aaa" />

        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID1</tp:OtherPartyActionBinding>
</tp:CanReceive>

<tp:CanReceive>
    <tp:ThisPartyActionBinding
        tp:id="companyA_ABID5"
        tp:action="data verifica"
        tp:packageId="CompanyA_ResponsePackage">
        <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true" />

        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID2</tp:OtherPartyActionBinding>
</tp:CanReceive>

<tp:CanReceive>
    <tp:ThisPartyActionBinding
        tp:id="companyA_ABID6"
        tp:action="esito verifica"
        tp:packageId="CompanyA_esitoPackage">

```

```
        <tp:BusinessTransactionCharacteristics
          tp:isNonRepudiationRequired="true"
          tp:isConfidential="transient"
          tp:isAuthenticated="persistent"
          tp:isTamperProof="none"
          tp:isAuthorizationRequired="true" />

        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
      </tp:ThisPartyActionBinding>

    <tp:OtherPartyActionBinding>companyB_ABID3</tp:OtherPartyActionBinding>
      </tp:CanReceive>
    </tp:ServiceBinding>
  </tp:CollaborationRole>

  <!-- Certificates used by the "richiedente" company -->
  <tp:Certificate
    tp:certId="CompanyA_AppCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyA_AppCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_SigningCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyA_SigningCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_EncryptionCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyA_EncryptionCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_ServerCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyA_ServerCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_ClientCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyA_ClientCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA1">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertA1_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA2">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertA2_Key</ds:KeyName>
```

```
</ds:KeyInfo>
</tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertA3">
  <ds:KeyInfo>
    <ds:KeyName>TrustedRootCertA3_Key</ds:KeyName>
  </ds:KeyInfo>
</tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertA4">
  <ds:KeyInfo>
    <ds:KeyName>TrustedRootCertA4_Key</ds:KeyName>
  </ds:KeyInfo>
</tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertA5">
  <ds:KeyInfo>
    <ds:KeyName>TrustedRootCertA5_Key</ds:KeyName>
  </ds:KeyInfo>
</tp:Certificate>

<tp:SecurityDetails
  tp:securityId="CompanyA_TransportSecurity">
  <tp:TrustAnchors>
    <tp:AnchorCertificateRef
      tp:certId="TrustedRootCertA1" />

    <tp:AnchorCertificateRef
      tp:certId="TrustedRootCertA2" />

    <tp:AnchorCertificateRef
      tp:certId="TrustedRootCertA4" />
  </tp:TrustAnchors>
</tp:SecurityDetails>

<tp:SecurityDetails
  tp:securityId="CompanyA_MessageSecurity">
  <tp:TrustAnchors>
    <tp:AnchorCertificateRef
      tp:certId="TrustedRootCertA3" />

    <tp:AnchorCertificateRef
      tp:certId="TrustedRootCertA5" />
  </tp:TrustAnchors>
</tp:SecurityDetails>

<tp:DeliveryChannel
  tp:channelId="asyncChannelA1"
  tp:transportId="transportA1"
  tp:docExchangeId="docExchangeA1">
  <tp:MessagingCharacteristics
    tp:syncReplyMode="none"
    tp:ackRequested="always"
    tp:duplicateElimination="always" />
</tp:DeliveryChannel>

<tp:Transport
  tp:transportId="transportA1">
  <tp:TransportSender>
    <tp:TransportProtocol
```

```

    tp:version="1.1">HTTP</tp:TransportProtocol>

    <tp:AccessAuthentication>basic</tp:AccessAuthentication>

    <tp:TransportClientSecurity>
      <tp:TransportSecurityProtocol
        tp:version="3.0">SSL</tp:TransportSecurityProtocol>

        <tp:ClientCertificateRef
          tp:certId="CompanyA_ClientCert" />

        <tp:ServerSecurityDetailsRef
          tp:securityId="CompanyA_TransportSecurity" />
      </tp:TransportClientSecurity>
    </tp:TransportSender>

    <tp:TransportReceiver>
      <tp:TransportProtocol
        tp:version="1.1">HTTP</tp:TransportProtocol>

        <tp:AccessAuthentication>basic</tp:AccessAuthentication>

        <tp:Endpoint
          tp:uri="https://www.landivivai.it/servlets/ebxmlhandler/async"
          tp:type="allPurpose" />

        <tp:TransportServerSecurity>
          <tp:TransportSecurityProtocol
            tp:version="3.0">SSL</tp:TransportSecurityProtocol>

            <tp:ServerCertificateRef
              tp:certId="CompanyA_ServerCert" />

            <tp:ClientSecurityDetailsRef
              tp:securityId="CompanyA_TransportSecurity" />
          </tp:TransportServerSecurity>
        </tp:TransportReceiver>
      </tp:Transport>

    <tp:DocExchange
      tp:docExchangeId="docExchangeA1">
      <tp:ebXMLSenderBinding
        tp:version="2.0">
        <tp:ReliableMessaging>
          <tp:Retries>3</tp:Retries>

          <tp:RetryInterval>PT2H</tp:RetryInterval>

        </tp:ReliableMessaging>
      </tp:ebXMLSenderBinding>
    </tp:DocExchange>

    <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
    </tp:ReliableMessaging>

    <tp:PersistDuration>P1D</tp:PersistDuration>

    <tp:SenderNonRepudiation>

    <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
ationProtocol>

    <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

```

```

<tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>

    <tp:SigningCertificateRef
      tp:certId="CompanyA_SigningCert" />
</tp:SenderNonRepudiation>

<tp:SenderDigitalEnvelope>
  <tp:DigitalEnvelopeProtocol
    tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

    <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

    <tp:EncryptionSecurityDetailsRef
      tp:securityId="CompanyA_MessageSecurity" />
  </tp:SenderDigitalEnvelope>
</tp:ebXMLSenderBinding>

<tp:ebXMLReceiverBinding
  tp:version="2.0">
    <tp:ReliableMessaging>
      <tp:Retries>3</tp:Retries>

      <tp:RetryInterval>PT2H</tp:RetryInterval>

    </tp:ReliableMessaging>
  </tp:ebXMLReceiverBinding>

<tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
</tp:ReliableMessaging>

<tp:PersistDuration>P1D</tp:PersistDuration>

<tp:ReceiverNonRepudiation>

<tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
ationProtocol>

<tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

<tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>

    <tp:SigningSecurityDetailsRef
      tp:securityId="CompanyA_MessageSecurity" />
</tp:ReceiverNonRepudiation>

<tp:ReceiverDigitalEnvelope>
  <tp:DigitalEnvelopeProtocol
    tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

    <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

    <tp:EncryptionCertificateRef
      tp:certId="CompanyA_EncryptionCert" />
  </tp:ReceiverDigitalEnvelope>
</tp:ebXMLReceiverBinding>
</tp:DocExchange>
</tp:PartyInfo>

<!-- Party info for CompanyB -->
  <tp:PartyInfo

```

```

    tp:partyName="coordinatore"
    tp:defaultMshChannelId="asyncChannelB1"
    tp:defaultMshPackageId="CompanyB_MshSignalPackage">
      <tp:PartyId
        tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-
type:duns">COORD01</tp:PartyId>

      <tp:PartyRef
        xlink:type="simple"
        xlink:href="http://cerere04.iet.unipi.it/about.html" />

      <tp:CollaborationRole>
        <tp:ProcessSpecification
          tp:version="1.2"
          tp:name="ingresso filiera"
          xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s1.xml"
          tp:uuid="urn:TraceService" />

        <tp:Role
          tp:name="coordinatore"
          xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s1.xml#coordinatore" />

        <tp:ApplicationCertificateRef
          tp:certId="CompanyB_AppCert" />

        <tp:ServiceBinding>
          <tp:Service>bpid:icann:rosettanet.org:3A4$2.0</tp:Service>

          <tp:CanSend>
            <tp:ThisPartyActionBinding
              tp:id="companyB_ABID1"
              tp:action="ack"
              tp:packageId="CompanyB_ReceiptAcknowledgmentPackage">
                <tp:BusinessTransactionCharacteristics
                  tp:isNonRepudiationRequired="true"
                  tp:isConfidential="transient"
                  tp:isAuthenticated="persistent"
                  tp:isTamperProof="none"
                  tp:isAuthorizationRequired="true"
                  tp:timeToAcknowledgeReceipt="PT2H" />

                <tp:ChannelId>asyncChannelB1</tp:ChannelId>
              </tp:ThisPartyActionBinding>

            <tp:OtherPartyActionBinding>companyA_ABID4</tp:OtherPartyActionBinding>
          </tp:CanSend>

          <tp:CanSend>
            <tp:ThisPartyActionBinding
              tp:id="companyB_ABID2"
              tp:action="data verifica"
              tp:packageId="CompanyB_ResponsePackage">
                <tp:BusinessTransactionCharacteristics
                  tp:isNonRepudiationRequired="true"
                  tp:isConfidential="transient"
                  tp:isAuthenticated="persistent"

```

```

        tp:isTamperProof="none"
        tp:isAuthorizationRequired="true"
        tp:timeToAcknowledgeReceipt="PT2H" />

        <tp:ActionContext
        tp:binaryCollaboration="richiesta ingresso"
        tp:businessTransactionActivity="richiesta ingresso"
        tp:requestOrResponseAction="aaa" />

        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyA_ABID5</tp:OtherPartyActionBinding>
    </tp:CanSend>

    <tp:CanSend>
        <tp:ThisPartyActionBinding
        tp:id="companyB_ABID3"
        tp:action="esito verifica"
        tp:packageId="CompanyB_esitoPackage">
            <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true" />

            <tp:ActionContext
            tp:binaryCollaboration="richiesta ingresso"
            tp:businessTransactionActivity="invio esito"
            tp:requestOrResponseAction="aaa" />

            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyA_ABID6</tp:OtherPartyActionBinding>
    </tp:CanSend>

    <tp:CanReceive>
        <tp:ThisPartyActionBinding
        tp:id="companyB_ABID4"
        tp:action="info richiesta"
        tp:packageId="CompanyB_RequestPackage">
            <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true"
            tp:timeToAcknowledgeReceipt="PT2H" />

            <tp:ActionContext
            tp:binaryCollaboration="richiesta ingresso"
            tp:businessTransactionActivity="richiesta ingresso"
            tp:requestOrResponseAction="aaa" />

            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyA_ABID1</tp:OtherPartyActionBinding>

```

```

        </tp:CanReceive>

        <tp:CanReceive>
            <tp:ThisPartyActionBinding
                tp:id="companyB_ABID5"
                tp:action="ack"
                tp:packageId="CompanyB_ReceiptAcknowledgmentPackage">
                <tp:BusinessTransactionCharacteristics
                    tp:isNonRepudiationRequired="true"
                    tp:isConfidential="transient"
                    tp:isAuthenticated="persistent"
                    tp:isTamperProof="none"
                    tp:isAuthorizationRequired="true" />

                <tp:ChannelId>asyncChannelB1</tp:ChannelId>
            </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>companyA_ABID2</tp:OtherPartyActionBinding>
    </tp:CanReceive>

    <tp:CanReceive>
        <tp:ThisPartyActionBinding
            tp:id="companyB_ABID6"
            tp:action="Purchase Order Request Action"
            tp:packageId="CompanyB_modPackage">
            <tp:BusinessTransactionCharacteristics
                tp:isNonRepudiationRequired="true"
                tp:isConfidential="transient"
                tp:isAuthenticated="persistent"
                tp:isTamperProof="none"
                tp:isAuthorizationRequired="true"
                tp:timeToAcknowledgeReceipt="PT2H" />

            <tp:ActionContext
                tp:binaryCollaboration="richiesta ingresso"
                tp:businessTransactionActivity="invio esito"
                tp:requestOrResponseAction="a" />

            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
        </tp:ThisPartyActionBinding>

    <tp:OtherPartyActionBinding>companyA_ABID3</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:ServiceBinding>
    </tp:CollaborationRole>

    <!-- Certificates used by the "Seller" company -->
    <tp:Certificate
        tp:certId="CompanyB_AppCert">
        <ds:KeyInfo>
            <ds:KeyName>CompanyB_AppCert_Key</ds:KeyName>
        </ds:KeyInfo>
    </tp:Certificate>

    <tp:Certificate
        tp:certId="CompanyB_SigningCert">
        <ds:KeyInfo>
            <ds:KeyName>CompanyB_Signingcert_Key</ds:KeyName>
        </ds:KeyInfo>
    </tp:Certificate>

```

```
<tp:Certificate
  tp:certId="CompanyB_EncryptionCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyB_EncryptionCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="CompanyB_ServerCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyB_ServerCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="CompanyB_ClientCert">
    <ds:KeyInfo>
      <ds:KeyName>CompanyB_ClientCert_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB4">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB4_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB5">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB5_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB6">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB6_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB7">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB7_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB8">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB8_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:SecurityDetails
  tp:securityId="CompanyB_TransportSecurity">
    <tp:TrustAnchors>
      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB5" />

      <tp:AnchorCertificateRef
```

```
        tp:certId="TrustedRootCertB6" />

        <tp:AnchorCertificateRef
            tp:certId="TrustedRootCertB4" />
    </tp:TrustAnchors>
</tp:SecurityDetails>

<tp:SecurityDetails
    tp:securityId="CompanyB_MessageSecurity">
    <tp:TrustAnchors>
        <tp:AnchorCertificateRef
            tp:certId="TrustedRootCertB8" />

        <tp:AnchorCertificateRef
            tp:certId="TrustedRootCertB7" />
    </tp:TrustAnchors>
</tp:SecurityDetails>

<!-- An asynchronous delivery channel -->
<tp:DeliveryChannel
    tp:channelId="asyncChannelB1"
    tp:transportId="transportB1"
    tp:docExchangeId="docExchangeB1">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="always"
        tp:duplicateElimination="always" />
</tp:DeliveryChannel>

<tp:Transport
    tp:transportId="transportB1">
    <tp:TransportSender>
        <tp:TransportProtocol
            tp:version="1.1">HTTP</tp:TransportProtocol>

        <tp:AccessAuthentication>basic</tp:AccessAuthentication>

        <tp:TransportClientSecurity>
            <tp:TransportSecurityProtocol
                tp:version="3.0">SSL</tp:TransportSecurityProtocol>

            <tp:ClientCertificateRef
                tp:certId="CompanyB_ClientCert" />

            <tp:ServerSecurityDetailsRef
                tp:securityId="CompanyB_TransportSecurity" />
        </tp:TransportClientSecurity>
    </tp:TransportSender>

    <tp:TransportReceiver>
        <tp:TransportProtocol
            tp:version="1.1">HTTP</tp:TransportProtocol>

        <tp:AccessAuthentication>basic</tp:AccessAuthentication>

        <tp:Endpoint
            tp:uri="https://cerere04.i.et.unipi.it:8080/servlets/"
            tp:type="allPurpose" />

        <tp:TransportServerSecurity>
            <tp:TransportSecurityProtocol
                tp:version="3.0">SSL</tp:TransportSecurityProtocol>
```

```

        <tp:ServerCertificateRef
          tp:certId="CompanyB_ServerCert" />

        <tp:ClientSecurityDetailsRef
          tp:securityId="CompanyB_TransportSecurity" />
      </tp:TransportServerSecurity>
    </tp:TransportReceiver>
  </tp:Transport>

  <tp:DocExchange
    tp:docExchangeId="docExchangeB1">
    <tp:ebXMLSenderBinding
      tp:version="2.0">
        <tp:ReliableMessaging>
          <tp:Retries>3</tp:Retries>

          <tp:RetryInterval>PT2H</tp:RetryInterval>

        </tp:ReliableMessaging>
      </tp:ebXMLSenderBinding>

      <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>

      <tp:PersistDuration>P1D</tp:PersistDuration>

      <tp:SenderNonRepudiation>

        <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
          ationProtocol>

        <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

        <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
          sha1</tp:SignatureAlgorithm>

        <tp:SigningCertificateRef
          tp:certId="CompanyB_SigningCert" />
      </tp:SenderNonRepudiation>

      <tp:SenderDigitalEnvelope>
        <tp:DigitalEnvelopeProtocol
          tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

        <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

        <tp:EncryptionSecurityDetailsRef
          tp:securityId="CompanyB_MessageSecurity" />
      </tp:SenderDigitalEnvelope>
    </tp:ebXMLSenderBinding>

    <tp:ebXMLReceiverBinding
      tp:version="2.0">
        <tp:ReliableMessaging>
          <tp:Retries>3</tp:Retries>

          <tp:RetryInterval>PT2H</tp:RetryInterval>

        </tp:ReliableMessaging>
      </tp:ebXMLReceiverBinding>

      <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>

      <tp:PersistDuration>P1D</tp:PersistDuration>
    </tp:DocExchange>
  </tp:DocExchange>

```

```
<tp:ReceiverNonRepudiation>

<tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
ationProtocol>

<tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

<tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>

    <tp:SigningSecurityDetailsRef
      tp:securityId="CompanyB_MessageSecurity" />
  </tp:ReceiverNonRepudiation>

  <tp:ReceiverDigitalEnvelope>
    <tp:DigitalEnvelopeProtocol
      tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

    <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

    <tp:EncryptionCertificateRef
      tp:certId="CompanyB_EncryptionCert" />
    </tp:ReceiverDigitalEnvelope>
  </tp:ebXMLReceiverBinding>
</tp:DocExchange>
</tp:PartyInfo>

<!-- SimplePart corresponding to the SOAP Envelope -->
<tp:SimplePart
  tp:id="CompanyA_MsgHdr"
  tp:mimetype="text/xml">
</tp:SimplePart>

<tp:SimplePart
  tp:id="CompanyB_MsgHdr"
  tp:mimetype="text/xml">
</tp:SimplePart>

<!-- SimplePart corresponding to a Receipt Acknowledgment business signal -
->
  <tp:SimplePart
    tp:id="CompanyA_ReceiptAcknowledgment"
    tp:mimetype="application/xml">
  </tp:SimplePart>

  <tp:SimplePart
    tp:id="CompanyB_ReceiptAcknowledgment"
    tp:mimetype="application/xml">
  </tp:SimplePart>

<!-- SimplePart corresponding to a request action -->
<tp:SimplePart
  tp:id="CompanyA_Request"
  tp:mimetype="application/xml">
</tp:SimplePart>

<tp:SimplePart
  tp:id="CompanyB_Request"
  tp:mimetype="application/xml">
</tp:SimplePart>
```



```
<!-- SimplePart corresponding to a response action -->
  <tp:SimplePart
    tp:id="CompanyA_Response"
    tp:mimetype="application/xml">
  </tp:SimplePart>

  <tp:SimplePart
    tp:id="CompanyB_Response"
    tp:mimetype="application/xml">
  </tp:SimplePart>

<!-- SimplePart corresponding to a invio esito action -->
  <tp:SimplePart
    tp:id="CompanyA_esito"
    tp:mimetype="application/xml">
  </tp:SimplePart>

  <tp:SimplePart
    tp:id="CompanyB_esito"
    tp:mimetype="application/xml">
  </tp:SimplePart>

<!-- SimplePart corresponding to a modifich e effettuate action -->
  <tp:SimplePart
    tp:id="CompanyA_mod"
    tp:mimetype="application/xml">
  </tp:SimplePart>

  <tp:SimplePart
    tp:id="CompanyB_mod"
    tp:mimetype="application/xml">
  </tp:SimplePart>

<!-- An ebXML message with a SOAP Envelope only -->
  <tp:Packaging
    tp:id="CompanyA_MshSignalPackage">
    <tp:ProcessingCapabilities
      tp:parse="true"
      tp:generate="true" />

    <tp:CompositeList>
      <tp:Composite
        tp:id="CompanyA_MshSignal"
        tp:mimetype="multipart/related"
        tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyA_MsgHdr" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

  <tp:Packaging
    tp:id="CompanyB_MshSignalPackage">
    <tp:ProcessingCapabilities
      tp:parse="true"
      tp:generate="true" />

    <tp:CompositeList>
      <tp:Composite
        tp:id="CompanyB_MshSignal"
        tp:mimetype="multipart/related"
        tp:mimeparameters="type=text/xml">
```

```
        <tp:Constituent
            tp:idref="CompanyB_MsgHdr" />
    </tp:Composite>
</tp:CompositeList>
</tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a request action payload --
>
    <tp:Packaging
        tp:id="CompanyA_RequestPackage">
        <tp:ProcessingCapabilities
            tp:parse="true"
            tp:generate="true" />

        <tp:CompositeList>
            <tp:Composite
                tp:id="CompanyA_RequestMsg"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyA_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyA_Request" />
                </tp:Composite>
            </tp:CompositeList>
        </tp:Packaging>

    <tp:Packaging
        tp:id="CompanyB_RequestPackage">
        <tp:ProcessingCapabilities
            tp:parse="true"
            tp:generate="true" />

        <tp:CompositeList>
            <tp:Composite
                tp:id="CompanyB_RequestMsg"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyB_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyB_Request" />
                </tp:Composite>
            </tp:CompositeList>
        </tp:Packaging>

    <!-- An ebXML message with a SOAP Envelope plus a response action payload -
    ->
    <tp:Packaging
        tp:id="CompanyA_ResponsePackage">
        <tp:ProcessingCapabilities
            tp:parse="true"
            tp:generate="true" />

        <tp:CompositeList>
            <tp:Composite
                tp:id="CompanyA_ResponseMsg"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyA_MsgHdr" />
```

```
        <tp:Constituent
            tp:idref="CompanyA_Response" />
    </tp:Composite>
</tp:CompositeList>
</tp:Packaging>

<tp:Packaging
    tp:id="CompanyB_ResponsePackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyB_ResponseMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyB_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyB_Response" />
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a invio esito action
payload -->
<tp:Packaging
    tp:id="CompanyA_esitoPackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyA_esitoMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyA_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyA_esito" />
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>

<tp:Packaging
    tp:id="CompanyB_esitoPackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyB_esitoMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyB_MsgHdr" />
```

```

        <tp:Constituent
            tp:idref="CompanyB_esito" />
    </tp:Composite>
</tp:CompositeList>
</tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a modifiché effectuée
action payload -->
<tp:Packaging
    tp:id="CompanyA_modPackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyA_modMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyA_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyA_mod" />
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>

<tp:Packaging
    tp:id="CompanyB_modPackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyB_modMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyB_MsgHdr" />

                <tp:Constituent
                    tp:idref="CompanyB_mod" />
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a Receipt Acknowledgment
payload -->
<tp:Packaging
    tp:id="CompanyA_ReceiptAcknowledgmentPackage">
    <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />

    <tp:CompositeList>
        <tp:Composite
            tp:id="CompanyA_ReceiptAcknowledgmentMsg"
            tp:mimetype="multipart/related"
            tp:mimeparameters="type=text/xml">
                <tp:Constituent
                    tp:idref="CompanyA_MsgHdr" />
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>

```

```
        <tp:Constituent
          tp:idref="CompanyA_ReceiptAcknowledgment" />
      </tp:Composite>
    </tp:CompositeList>
  </tp:Packaging>

  <tp:Packaging
    tp:id="CompanyB_ReceiptAcknowledgmentPackage">
    <tp:ProcessingCapabilities
      tp:parse="true"
      tp:generate="true" />

    <tp:CompositeList>
      <tp:Composite
        tp:id="CompanyB_ReceiptAcknowledgmentMsg"
        tp:mimetype="multipart/related"
        tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyB_MsgHdr" />

          <tp:Constituent
            tp:idref="CompanyB_ReceiptAcknowledgment" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

    <tp:Comment
      xml:lang="en-US">Accordo per l'ingresso in filiera tra Vivai Landi e
      Coordinatore</tp:Comment>
  </tp:CollaborationProtocolAgreement>
```

Appendice I

CPA per l'ordine delle piantine

```
<?xml version="1.0"?>
<!-- Copyright UN/CEFACT and OASIS, 2001. All Rights Reserved. -->
<tp:CollaborationProtocolAgreement
  xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-
2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd cpp-cpa-2_0a.xsd"
  tp:cpaid="uri:acquistopiantecpa"
  tp:version="2_0a">
  <tp:Status
    tp:value="signed" />

  <tp:Start>2005-05-20T07:21:00Z</tp:Start>

  <tp:End>2007-05-20T07:21:00Z</tp:End>

  <tp:ConversationConstraints
    tp:invocationLimit="100"
    tp:concurrentConversations="10" />

  <!-- Party info for CompanyA -->
  <tp:PartyInfo
    tp:partyName="Azienda Agricola Maestrale"
    tp:defaultMshChannelId="asyncChannelA1"
    tp:defaultMshPackageId="CompanyA_MshSignalPackage">
    <tp:PartyId>AM00002</tp:PartyId>

    <tp:PartyRef
      xlink:href="http://www.maestrale.it/info.html" />

    <tp:CollaborationRole>
      <tp:ProcessSpecification
        tp:version="1.2"
        tp:name="acquisto piante"
        xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s2.xml"
        tp:uuid="urn:acquistopiante" />

      <tp:Role
        tp:name="acquirente"
        xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s2.xml#acquirente" />

      <tp:ApplicationCertificateRef
```

```
tp:certId="CompanyA_AppCert" />

<tp:ServiceBinding>
  <tp:Service>bpid:icann:iet.unipi.it:3A5$2.0</tp:Service>

  <tp:CanSend>
    <tp:ThisPartyActionBinding
      tp:id="companyA_ABID1"
      tp:action="ordine"
      tp:packageId="CompanyA_RequestPackage">
      <tp:BusinessTransactionCharacteristics
        tp:isNonRepudiationRequired="true"
        tp:isConfidential="transient"
        tp:isAuthenticated="persistent"
        tp:isTamperProof="none"
        tp:isAuthorizationRequired="true"
        tp:timeToAcknowledgeReceipt="PT2H" />

      <tp:ActionContext
        tp:binaryCollaboration="acquisto piante"
        tp:businessTransactionActivity="ordine acquisto"
        tp:requestOrResponseAction="req" />

      <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

  <tp:OtherPartyActionBinding>companyB_ABID3</tp:OtherPartyActionBinding>
</tp:CanSend>

<tp:CanSend>
  <tp:ThisPartyActionBinding
    tp:id="companyA_ABID2"
    tp:action="ack"
    tp:packageId="CompanyA_ReceiptAcknowledgmentPackage">
    <tp:BusinessTransactionCharacteristics
      tp:isNonRepudiationRequired="true"
      tp:isConfidential="transient"
      tp:isAuthenticated="persistent"
      tp:isTamperProof="none"
      tp:isAuthorizationRequired="true" />

    <tp:ChannelId>asyncChannelA1</tp:ChannelId>

    <tp:ActionContext
      tp:binaryCollaboration="acquisto piante"
      tp:businessTransactionActivity="ordine acquisto"
      tp:requestOrResponseAction="res" />
  </tp:ThisPartyActionBinding>

  <tp:OtherPartyActionBinding>companyB_ABID4</tp:OtherPartyActionBinding>
</tp:CanSend>

<tp:CanSend>
  <tp:ThisPartyActionBinding
    tp:id="companyA_ABID3"
    tp:action="info pagamento"
    tp:packageId="CompanyA_infoPackage">
    <tp:BusinessTransactionCharacteristics
      tp:isNonRepudiationRequired="true"
      tp:isConfidential="transient"
      tp:isAuthenticated="persistent"
```

```

        tp:isTamperProof="persistent"
        tp:isAuthorizationRequired="true"
        tp:timeToAcknowledgeReceipt="PT4H" />

        <tp:ActionContext
            tp:binaryCollaboration="acquisto piante"
            tp:businessTransactionActivity="invio pagamento"
            tp:requestOrResponseAction="req" />

        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID5</tp:OtherPartyActionBinding>
    <tp:CanSend>

        <tp:CanReceive>
            <tp:ThisPartyActionBinding
                tp:id="companyA_ABID4"
                tp:action="ack"
                tp:packageId="CompanyA_ReceiptAcknowledgmentPackage">
                <tp:BusinessTransactionCharacteristics
                    tp:isNonRepudiationRequired="true"
                    tp:isConfidential="transient"
                    tp:isAuthenticated="persistent"
                    tp:isTamperProof="persistent"
                    tp:isAuthorizationRequired="true"
                    tp:timeToAcknowledgeReceipt="PT2H" />

                <tp:ActionContext
                    tp:binaryCollaboration="acquisto piante"
                    tp:businessTransactionActivity="ordine acquisto"
                    tp:requestOrResponseAction="req" />

                <tp:ChannelId>asyncChannelA1</tp:ChannelId>
            </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID1</tp:OtherPartyActionBinding>
    <tp:CanReceive>

        <tp:CanReceive>
            <tp:ThisPartyActionBinding
                tp:id="companyA_ABID5"
                tp:action="esito ordine"
                tp:packageId="CompanyA_ResponsePackage">
                <tp:BusinessTransactionCharacteristics
                    tp:isNonRepudiationRequired="true"
                    tp:isConfidential="transient"
                    tp:isAuthenticated="persistent"
                    tp:isTamperProof="none"
                    tp:isAuthorizationRequired="true" />

                <tp:ChannelId>asyncChannelA1</tp:ChannelId>
            </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyB_ABID2</tp:OtherPartyActionBinding>
    <tp:CanReceive>

        </tp:ServiceBinding>
    </tp:CollaborationRole>

```



```
<!-- Certificates used by the "richiedente" company -->
  <tp:Certificate
    tp:certId="CompanyA_AppCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_AppCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_SigningCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_SigningCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_EncryptionCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_EncryptionCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_ServerCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_ServerCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="CompanyA_ClientCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyA_ClientCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA1">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA1_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA2">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA2_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA3">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA3_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

  <tp:Certificate
    tp:certId="TrustedRootCertA4">
      <ds:KeyInfo>
        <ds:KeyName>TrustedRootCertA4_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>
```

```
<tp:Certificate
  tp:certId="TrustedRootCertA5">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertA5_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:SecurityDetails
  tp:securityId="CompanyA_TransportSecurity">
    <tp:TrustAnchors>
      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertA1" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertA2" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertA4" />
    </tp:TrustAnchors>
  </tp:SecurityDetails>

<tp:SecurityDetails
  tp:securityId="CompanyA_MessageSecurity">
    <tp:TrustAnchors>
      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertA3" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertA5" />
    </tp:TrustAnchors>
  </tp:SecurityDetails>

<tp:DeliveryChannel
  tp:channelId="asyncChannelA1"
  tp:transportId="transportA1"
  tp:docExchangeId="docExchangeA1">
    <tp:MessagingCharacteristics
      tp:syncReplyMode="none"
      tp:ackRequested="always"
      tp:duplicateElimination="always" />
  </tp:DeliveryChannel>

<tp:Transport
  tp:transportId="transportA1">
    <tp:TransportSender>
      <tp:TransportProtocol
        tp:version="1.1">HTTP</tp:TransportProtocol>

      <tp:AccessAuthentication>basic</tp:AccessAuthentication>

      <tp:TransportClientSecurity>
        <tp:TransportSecurityProtocol
          tp:version="3.0">SSL</tp:TransportSecurityProtocol>

        <tp:ClientCertificateRef
          tp:certId="CompanyA_ClientCert" />

        <tp:ServerSecurityDetailsRef
          tp:securityId="CompanyA_TransportSecurity" />
      </tp:TransportClientSecurity>
    </tp:TransportSender>
```

```
<tp:TransportReceiver>
  <tp:TransportProtocol
    tp:version="1.1">HTTP</tp:TransportProtocol>

    <tp:AccessAuthentication>basic</tp:AccessAuthentication>

    <tp:Endpoint
      tp:uri="https://www.maestrale.it:8080/servlets/"
      tp:type="allPurpose" />

    <tp:TransportServerSecurity>
      <tp:TransportSecurityProtocol
        tp:version="3.0">SSL</tp:TransportSecurityProtocol>

      <tp:ServerCertificateRef
        tp:certId="CompanyA_ServerCert" />

      <tp:ClientSecurityDetailsRef
        tp:securityId="CompanyA_TransportSecurity" />
    </tp:TransportServerSecurity>
  </tp:TransportReceiver>
</tp:Transport>

<tp:DocExchange
  tp:docExchangeId="docExchangeA1">
  <tp:ebXMLSenderBinding
    tp:version="2.0">
    <tp:ReliableMessaging>
      <tp:Retries>3</tp:Retries>

      <tp:RetryInterval>PT2H</tp:RetryInterval>

    <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
    </tp:ReliableMessaging>

    <tp:PersistDuration>P1D</tp:PersistDuration>

    <tp:SenderNonRepudiation>

    <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
ationProtocol>

    <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

    <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>

      <tp:SigningCertificateRef
        tp:certId="CompanyA_SigningCert" />
    </tp:SenderNonRepudiation>

    <tp:SenderDigitalEnvelope>
      <tp:DigitalEnvelopeProtocol
        tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

      <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

      <tp:EncryptionSecurityDetailsRef
        tp:securityId="CompanyA_MessageSecurity" />
    </tp:SenderDigitalEnvelope>
```

```

        </tp:ebXMLSenderBinding>

        <tp:ebXMLReceiverBinding
            tp:version="2.0">
            <tp:ReliableMessaging>
                <tp:Retries>3</tp:Retries>

                <tp:RetryInterval>PT2H</tp:RetryInterval>

            <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
            </tp:ReliableMessaging>

            <tp:PersistDuration>P1D</tp:PersistDuration>

            <tp:ReceiverNonRepudiation>

            <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
            ationProtocol>

            <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

            <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
            sha1</tp:SignatureAlgorithm>

                <tp:SigningSecurityDetailsRef
                    tp:securityId="CompanyA_MessageSecurity" />
            </tp:ReceiverNonRepudiation>

            <tp:ReceiverDigitalEnvelope>
                <tp:DigitalEnvelopeProtocol
                    tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

                <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

                <tp:EncryptionCertificateRef
                    tp:certId="CompanyA_EncryptionCert" />
            </tp:ReceiverDigitalEnvelope>
        </tp:ebXMLReceiverBinding>
    </tp:DocExchange>
</tp:PartyInfo>

<!-- Party info for CompanyB -->
    <tp:PartyInfo
        tp:partyName="Vivai Landi"
        tp:defaultMshChannelId="asyncChannelB1"
        tp:defaultMshPackageId="CompanyB_MshSignalPackage">
        <tp:PartyId>VL00001</tp:PartyId>

        <tp:PartyRef
            xlink:type="simple"
            xlink:href="http://www.landivivai.it/info.html" />

        <tp:CollaborationRole>
            <tp:ProcessSpecification
                tp:version="1.2"
                tp:name="acquisto piante"
                xlink:type="simple"

            xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
            s2.xml"

```

```
        tp:uuid="urn:TraceService" />

    <tp:Role
      tp:name="venditore"
      xlink:type="simple"

xlink:href="http://cerere04.iet.unipi.it:8082/omar/processes/businessproces
s2.xml#venditore" />

    <tp:ApplicationCertificateRef
      tp:certId="CompanyB_AppCert" />

    <tp:ServiceBinding>
      <tp:Service>bpid:icann:rosettanet.org:3A4$2.0</tp:Service>

      <tp:CanSend>
        <tp:ThisPartyActionBinding
          tp:id="companyB_ABID1"
          tp:action="ack"
          tp:packageId="CompanyB_ReceiptAcknowledgmentPackage">
          <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true"
            tp:timeToAcknowledgeReceipt="PT2H" />

            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
          </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>companyA_ABID4</tp:OtherPartyActionBinding>
      </tp:CanSend>

      <tp:CanSend>
        <tp:ThisPartyActionBinding
          tp:id="companyB_ABID2"
          tp:action="esito ordine"
          tp:packageId="CompanyB_ResponsePackage">
          <tp:BusinessTransactionCharacteristics
            tp:isNonRepudiationRequired="true"
            tp:isConfidential="transient"
            tp:isAuthenticated="persistent"
            tp:isTamperProof="none"
            tp:isAuthorizationRequired="true"
            tp:timeToAcknowledgeReceipt="PT2H" />

          <tp:ActionContext
            tp:binaryCollaboration="acquisto piante"
            tp:businessTransactionActivity="ordine acquisto"
            tp:requestOrResponseAction="res" />

          <tp:ChannelId>asyncChannelB1</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>companyA_ABID5</tp:OtherPartyActionBinding>
      </tp:CanSend>

      <tp:CanReceive>
        <tp:ThisPartyActionBinding
          tp:id="companyB_ABID3"
```

```
        tp:action="ordine"
        tp:packageId="CompanyB_RequestPackage">
        <tp:BusinessTransactionCharacteristics
        tp:isNonRepudiationRequired="true"
        tp:isConfidential="transient"
        tp:isAuthenticated="persistent"
        tp:isTamperProof="none"
        tp:isAuthorizationRequired="true"
        tp:timeToAcknowledgeReceipt="PT2H" />

        <tp:ActionContext
        tp:binaryCollaboration="Acquisto piante"
        tp:businessTransactionActivity="ordine acquisto"
        tp:requestOrResponseAction="req" />

        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyA_ABID1</tp:OtherPartyActionBinding>
    </tp:CanReceive>

    <tp:CanReceive>
    <tp:ThisPartyActionBinding
    tp:id="companyB_ABID4"
    tp:action="ack"
    tp:packageId="CompanyB_ReceiptAcknowledgmentPackage">
    <tp:BusinessTransactionCharacteristics
    tp:isNonRepudiationRequired="true"
    tp:isConfidential="transient"
    tp:isAuthenticated="persistent"
    tp:isTamperProof="none"
    tp:isAuthorizationRequired="true" />

    <tp:ChannelId>asyncChannelB1</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>companyA_ABID2</tp:OtherPartyActionBinding>
    </tp:CanReceive>

    <tp:CanReceive>
    <tp:ThisPartyActionBinding
    tp:id="companyB_ABID5"
    tp:action="info pagamento"
    tp:packageId="CompanyB_infoPackage">
    <tp:BusinessTransactionCharacteristics
    tp:isNonRepudiationRequired="true"
    tp:isConfidential="transient"
    tp:isAuthenticated="persistent"
    tp:isTamperProof="none"
    tp:isAuthorizationRequired="true"
    tp:timeToAcknowledgeReceipt="PT2H" />

    <tp:ActionContext
    tp:binaryCollaboration="acquisto piante"
    tp:businessTransactionActivity="invio pagamento"
    tp:requestOrResponseAction="req" />

    <tp:ChannelId>asyncChannelB1</tp:ChannelId>
    </tp:ThisPartyActionBinding>
```

```
<tp:OtherPartyActionBinding>companyA_ABID3</tp:OtherPartyActionBinding>
    </tp:CanReceive>
  </tp:ServiceBinding>
</tp:CollaborationRole>

<!-- Certificates used by the "Seller" company -->
  <tp:Certificate
    tp:certId="CompanyB_AppCert">
      <ds:KeyInfo>
        <ds:KeyName>CompanyB_AppCert_Key</ds:KeyName>
      </ds:KeyInfo>
    </tp:Certificate>

    <tp:Certificate
      tp:certId="CompanyB_SigningCert">
        <ds:KeyInfo>
          <ds:KeyName>CompanyB_Signingcert_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="CompanyB_EncryptionCert">
        <ds:KeyInfo>
          <ds:KeyName>CompanyB_EncryptionCert_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="CompanyB_ServerCert">
        <ds:KeyInfo>
          <ds:KeyName>CompanyB_ServerCert_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="CompanyB_ClientCert">
        <ds:KeyInfo>
          <ds:KeyName>CompanyB_ClientCert_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="TrustedRootCertB4">
        <ds:KeyInfo>
          <ds:KeyName>TrustedRootCertB4_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="TrustedRootCertB5">
        <ds:KeyInfo>
          <ds:KeyName>TrustedRootCertB5_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>

    <tp:Certificate
      tp:certId="TrustedRootCertB6">
        <ds:KeyInfo>
          <ds:KeyName>TrustedRootCertB6_Key</ds:KeyName>
        </ds:KeyInfo>
      </tp:Certificate>
```

```
<tp:Certificate
  tp:certId="TrustedRootCertB7">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB7_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:Certificate
  tp:certId="TrustedRootCertB8">
    <ds:KeyInfo>
      <ds:KeyName>TrustedRootCertB8_Key</ds:KeyName>
    </ds:KeyInfo>
  </tp:Certificate>

<tp:SecurityDetails
  tp:securityId="CompanyB_TransportSecurity">
    <tp:TrustAnchors>
      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB5" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB6" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB4" />
    </tp:TrustAnchors>
  </tp:SecurityDetails>

<tp:SecurityDetails
  tp:securityId="CompanyB_MessageSecurity">
    <tp:TrustAnchors>
      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB8" />

      <tp:AnchorCertificateRef
        tp:certId="TrustedRootCertB7" />
    </tp:TrustAnchors>
  </tp:SecurityDetails>

<!-- An asynchronous delivery channel -->
<tp:DeliveryChannel
  tp:channelId="asyncChannelB1"
  tp:transportId="transportB1"
  tp:docExchangeId="docExchangeB1">
  <tp:MessagingCharacteristics
    tp:syncReplyMode="none"
    tp:ackRequested="always"
    tp:duplicateElimination="always" />
  </tp:DeliveryChannel>

<tp:Transport
  tp:transportId="transportB1">
  <tp:TransportSender>
    <tp:TransportProtocol
      tp:version="1.1">HTTP</tp:TransportProtocol>

    <tp:AccessAuthentication>basic</tp:AccessAuthentication>

    <tp:TransportClientSecurity>
      <tp:TransportSecurityProtocol
        tp:version="3.0">SSL</tp:TransportSecurityProtocol>

      <tp:ClientCertificateRef
```



```
        tp:certId="CompanyB_ClientCert" />

        <tp:ServerSecurityDetailsRef
          tp:securityId="CompanyB_TransportSecurity" />
      </tp:TransportClientSecurity>
</tp:TransportSender>

<tp:TransportReceiver>
  <tp:TransportProtocol
    tp:version="1.1">HTTP</tp:TransportProtocol>

    <tp:AccessAuthentication>basic</tp:AccessAuthentication>

    <tp:Endpoint
      tp:uri="https://www.landivivai.it/servlets/ebxmlhandler/async"
      tp:type="allPurpose" />

    <tp:TransportServerSecurity>
      <tp:TransportSecurityProtocol
        tp:version="3.0">SSL</tp:TransportSecurityProtocol>

      <tp:ServerCertificateRef
        tp:certId="CompanyB_ServerCert" />

      <tp:ClientSecurityDetailsRef
        tp:securityId="CompanyB_TransportSecurity" />
    </tp:TransportServerSecurity>
  </tp:TransportReceiver>
</tp:Transport>

<tp:DocExchange
  tp:docExchangeId="docExchangeB1">
  <tp:ebXMLSenderBinding
    tp:version="2.0">
    <tp:ReliableMessaging>
      <tp:Retries>3</tp:Retries>

      <tp:RetryInterval>PT2H</tp:RetryInterval>

    </tp:ReliableMessaging>
  </tp:ebXMLSenderBinding>
</tp:DocExchange>

<tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
</tp:ReliableMessaging>

<tp:PersistDuration>P1D</tp:PersistDuration>

<tp:SenderNonRepudiation>

  <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi-
ationProtocol>

  <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

  <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>

  <tp:SigningCertificateRef
    tp:certId="CompanyB_SigningCert" />
</tp:SenderNonRepudiation>

<tp:SenderDigitalEnvelope>
  <tp:DigitalEnvelopeProtocol
```

```

        tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

        <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

        <tp:EncryptionSecurityDetailsRef
          tp:securityId="CompanyB_MessageSecurity" />
      </tp:SenderDigitalEnvelope>
    </tp:ebXMLSenderBinding>

    <tp:ebXMLReceiverBinding
      tp:version="2.0">
        <tp:ReliableMessaging>
          <tp:Retries>3</tp:Retries>

          <tp:RetryInterval>PT2H</tp:RetryInterval>

        <tp:MessageOrderSemantics>Guaranteed</tp:MessageOrderSemantics>
        </tp:ReliableMessaging>

        <tp:PersistDuration>P1D</tp:PersistDuration>

        <tp:ReceiverNonRepudiation>

        <tp:NonRepudiationProtocol>http://www.w3.org/2000/09/xmldsig#</tp:NonRepudi
          ationProtocol>

        <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

        <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
          sha1</tp:SignatureAlgorithm>

        <tp:SigningSecurityDetailsRef
          tp:securityId="CompanyB_MessageSecurity" />
      </tp:ReceiverNonRepudiation>

      <tp:ReceiverDigitalEnvelope>
        <tp:DigitalEnvelopeProtocol
          tp:version="2.0">S/MIME</tp:DigitalEnvelopeProtocol>

        <tp:EncryptionAlgorithm>DES-CBC</tp:EncryptionAlgorithm>

        <tp:EncryptionCertificateRef
          tp:certId="CompanyB_EncryptionCert" />
      </tp:ReceiverDigitalEnvelope>
    </tp:ebXMLReceiverBinding>
  </tp:DocExchange>
</tp:PartyInfo>

<!-- SimplePart corresponding to the SOAP Envelope -->
  <tp:SimplePart
    tp:id="CompanyA_MsgHdr"
    tp:mimetype="text/xml">
  </tp:SimplePart>

  <tp:SimplePart
    tp:id="CompanyB_MsgHdr"
    tp:mimetype="text/xml">
  </tp:SimplePart>

```

```
<!-- SimplePart corresponding to a Receipt Acknowledgment business signal -
->
    <tp:SimplePart
      tp:id="CompanyA_ReceiptAcknowledgment"
      tp:mimetype="application/xml">
    </tp:SimplePart>

    <tp:SimplePart
      tp:id="CompanyB_ReceiptAcknowledgment"
      tp:mimetype="application/xml">
    </tp:SimplePart>

<!-- SimplePart corresponding to a request action -->
    <tp:SimplePart
      tp:id="CompanyA_Ordine"
      tp:mimetype="application/xml">
    </tp:SimplePart>

    <tp:SimplePart
      tp:id="CompanyB_Ordine"
      tp:mimetype="application/xml">
    </tp:SimplePart>

<!-- SimplePart corresponding to a response action -->
    <tp:SimplePart
      tp:id="CompanyA_Response"
      tp:mimetype="application/xml">
    </tp:SimplePart>

    <tp:SimplePart
      tp:id="CompanyB_Response"
      tp:mimetype="application/xml">
    </tp:SimplePart>

<!-- SimplePart corresponding to a invio esito action -->
    <tp:SimplePart
      tp:id="CompanyA_noteconsegna"
      tp:mimetype="application/xml">
    </tp:SimplePart>

    <tp:SimplePart
      tp:id="CompanyB_noteconsegna"
      tp:mimetype="application/xml">
    </tp:SimplePart>

<!-- SimplePart corresponding to a modifiche effettuate action -->
    <tp:SimplePart
      tp:id="CompanyA_info"
      tp:mimetype="application/xml">
    </tp:SimplePart>

    <tp:SimplePart
      tp:id="CompanyB_info"
      tp:mimetype="application/xml">
    </tp:SimplePart>

<!-- An ebXML message with a SOAP Envelope only -->
    <tp:Packaging
      tp:id="CompanyA_MshSignalPackage">
      <tp:ProcessingCapabilities
        tp:parse="true"
        tp:generate="true" />
```

```
<tp:CompositeList>
  <tp:Composite
    tp:id="CompanyA_MshSignal"
    tp:mimetype="multipart/related"
    tp:mimeparameters="type=text/xml">
    <tp:Constituent
      tp:idref="CompanyA_MsgHdr" />
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>

<tp:Packaging
  tp:id="CompanyB_MshSignalPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyB_MshSignal"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyB_MsgHdr" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a request action payload --
>
<tp:Packaging
  tp:id="CompanyA_RequestPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyA_RequestMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyA_MsgHdr" />

        <tp:Constituent
          tp:idref="CompanyA_Ordine" />
        <tp:Constituent
          tp:idref="CompanyA_noteconsegna" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

<tp:Packaging
  tp:id="CompanyB_RequestPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyB_RequestMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
```

```
<tp:Constituent
  tp:idref="CompanyB_MsgHdr" />

  <tp:Constituent
    tp:idref="CompanyB_Ordine" />
  <tp:Constituent
    tp:idref="CompanyB_noteconsegna" />
</tp:Composite>
</tp:CompositeList>
</tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a response action payload -
-->
<tp:Packaging
  tp:id="CompanyA_ResponsePackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyA_ResponseMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
      <tp:Constituent
        tp:idref="CompanyA_MsgHdr" />

        <tp:Constituent
          tp:idref="CompanyA_Response" />
      </tp:Composite>
    </tp:CompositeList>
  </tp:Packaging>

<tp:Packaging
  tp:id="CompanyB_ResponsePackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyB_ResponseMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
      <tp:Constituent
        tp:idref="CompanyB_MsgHdr" />

        <tp:Constituent
          tp:idref="CompanyB_Response" />
      </tp:Composite>
    </tp:CompositeList>
  </tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a modifichie effettuate
action payload -->
<tp:Packaging
  tp:id="CompanyA_infoPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
```

```
<tp:Composite
  tp:id="CompanyA_infoMsg"
  tp:mimetype="multipart/related"
  tp:mimeparameters="type=text/xml">
  <tp:Constituent
    tp:idref="CompanyA_MsgHdr" />

    <tp:Constituent
      tp:idref="CompanyA_info" />
  </tp:Composite>
</tp:CompositeList>
</tp:Packaging>

<tp:Packaging
  tp:id="CompanyB_modPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyB_infoMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyB_MsgHdr" />

          <tp:Constituent
            tp:idref="CompanyB_info" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

<!-- An ebXML message with a SOAP Envelope plus a Receipt Acknowledgment
payload -->
<tp:Packaging
  tp:id="CompanyA_ReceiptAcknowledgmentPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
      tp:id="CompanyA_ReceiptAcknowledgmentMsg"
      tp:mimetype="multipart/related"
      tp:mimeparameters="type=text/xml">
        <tp:Constituent
          tp:idref="CompanyA_MsgHdr" />

          <tp:Constituent
            tp:idref="CompanyA_ReceiptAcknowledgment" />
        </tp:Composite>
      </tp:CompositeList>
    </tp:Packaging>

<tp:Packaging
  tp:id="CompanyB_ReceiptAcknowledgmentPackage">
  <tp:ProcessingCapabilities
    tp:parse="true"
    tp:generate="true" />

  <tp:CompositeList>
    <tp:Composite
```

```
    tp:id="CompanyB_ReceiptAcknowledgmentMsg"
    tp:mimetype="multipart/related"
    tp:mimeparameters="type=text/xml">
      <tp:Constituent
        tp:idref="CompanyB_MsgHdr" />

      <tp:Constituent
        tp:idref="CompanyB_ReceiptAcknowledgment" />
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>

  <tp:Comment
    xml:lang="en-US">Accordo di compravendita tra Azienda Agricola
    Maestrone e Vivai Landi</tp:Comment>
</tp:CollaborationProtocolAgreement>
```

Appendice L

Files per la realizzazione di un Plug-in ebMail

PLUGIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<Plugin id="sample" version="1.0">
  <Name>Sample Plugin</Name>

  <Description>semplice plugin per l'acquisto di piante di
  cavolo</Description>

  <EbmailDocuments>
    <Document>
      <Name>ordine</Name>

      <Description>Semplice form per l'ordine di cavoli</Description>

      <DataClass>OrderDocumentData</DataClass>

      <HandlerClass>OrderDocumentHandler</HandlerClass>

      <HandlesEbxmlMessages>
        <MessageHeader>
          <CPAId>CPAID</CPAId>

          <Service>ordine</Service>

          <Action>ordine</Action>
        </MessageHeader>
      </HandlesEbxmlMessages>

      <SupportsEmailActions>
        <New />

        <NewFrom />
      </SupportsEmailActions>
    </Document>
  </EbmailDocuments>

  <Properties>
  </Properties>
</Plugin>
```

ORDERPANE.JAVA

```
import javax.swing.JTextField;
import javax.swing.JLabel;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;

public class OrderPane extends JPanel {
```



```
//dichiarazione delle variabili membro della classe
private JLabel dataLab, numeroLab;

private JTextField data, numero;

//costruttore
public OrderPane() {
    dataLab = new JLabel("Data massima accettabile per l'invio:");
    data = new JTextField(10);
    numeroLab = new JLabel("Numero di Piante richieste:");
    numero = new JTextField(5);
    layoutUI();
}

private void layoutUI() {
    GridBagLayout gridbag = new GridBagLayout();
    setLayout(gridbag);
    GridBagConstraints c = new GridBagConstraints();
    c.gridx = 0;
    c.gridy = 0;
    c.gridwidth = 1;
    c.gridheight = 1;
    c.weightx = 0;
    c.weighty = 0;
    c.anchor = GridBagConstraints.WEST;
    c.fill = GridBagConstraints.NONE;
    add(dataLab, c);
    c.gridx = 1;
    c.gridy = 0;
    c.gridwidth = 1;
    c.gridheight = 1;
    c.weightx = 1;
    c.weighty = 0;
    c.anchor = GridBagConstraints.WEST;
    c.fill = GridBagConstraints.NONE;
    add(data, c);
    c.gridx = 0;
    c.gridy = 1;
    c.gridwidth = 1;
    c.gridheight = 1;
    c.weightx = 0;
    c.weighty = 0;
    c.anchor = GridBagConstraints.WEST;
    c.fill = GridBagConstraints.NONE;
    add(numeroLab, c);
    c.gridx = 1;
    c.gridy = 1;
    c.gridwidth = 1;
    c.gridheight = 1;
    c.weightx = 1;
    c.weighty = 0;
    c.anchor = GridBagConstraints.WEST;
    c.fill = GridBagConstraints.NONE;
    add(numero, c);
}

//ritorna la data
public String getData() {
    return data.getText();
}

//setta se editabile o no
public void setEditable(boolean edit) {
```

```
        data.setEditable(edit);
        numero.setEditable(edit);
    }

    //ritorna il numero di piante
    public int getNumero() throws NumberFormatException {
        return Integer.parseInt(numero.getText());
    }

    //setta la data
    public void setData(String add) {
        data.setText(add);
    }

    //setta il numero di piante
    public void setNumero(int num) {
        if ( num > 0 ) {
            numero.setText(Integer.toString(num));
        }
        else {
            numero.setText("");
        }
    }
}
```

ORDERDOCUMENTDATA.JAVA

```
import java.io.IOException;
import java.io.InputStream;
import java.io.ByteArrayOutputStream;
import javax.activation.DataHandler;
import javax.xml.soap.SOAPException;
import javax.xml.*;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.input.SAXBuilder;
import org.jdom.output.XMLOutputter;
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentData;
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentException;
import hk.hku.cecid.phoenix.ebmail.util.ByteArrayDataSource;
import hk.hku.cecid.phoenix.message.packaging.AttachmentDataSource;
import hk.hku.cecid.phoenix.message.packaging.EbxmlMessage;
import hk.hku.cecid.phoenix.message.packaging.MessageHeader;
import hk.hku.cecid.phoenix.message.packaging.PayloadContainer;

public class OrderDocumentData implements DocumentData {
    //campi relativi al messaggio ebxml
    private static final String contentId = "0";

    private static final String cpa = "CPAID";

    private static final String service = "ordine";

    private static final String action = "ordine";

    //campi messaggio xml
    private static final String ordineName = "ordine";

    private static final String dataName = "data";
```

```
private static final String pianteName = "piante";

//campi privati classe
private String data;

private int numero = - 1;

public void load(DocumentData sourceDocument) throws DocumentException {
    if ( sourceDocument instanceof OrderDocumentData ) {
        OrderDocumentData doc = (OrderDocumentData)sourceDocument;
        data = doc.getData();
        numero = doc.getNumero();
    }
}

public void load(EbxmlMessage message) throws DocumentException {
    DataHandler handler = null;
    PayloadContainer container = message.getPayloadContainer(contentId);
    if ( container == null ) {
        throw new DocumentException("Invalid message format:" + "It doesn't
contain required payload");
    }
    try {
        handler = container.getDataHandler();
    }
    catch (SOAPException e) {
        throw new DocumentException("SoapException occur:" + e.toString());
    }
    InputStream stream = null;
    try {
        stream = handler.getDataSource().getInputStream();
    }
    catch (IOException e) {
        throw new DocumentException("Unexpected I/O exception occur:" +
e.toString());
    }
    SAXBuilder builder = new SAXBuilder(false);
    Document doc = null;
    try {
        doc = builder.build(stream);
    }
    catch (JDOMException e) {
        throw new DocumentException("Invalid message format:" +
e.toString());
    }
    Element ordineNode = doc.getRootElement();
    if ( ordineName.equals(ordineNode.getName()) ) {
        Element dataNode = ordineNode.getChild(dataName);
        if ( dataNode != null ) {
            data = dataNode.getText();
            if ( data == null || data.length() == 0 ) {
                throw new DocumentException("Invalid message format");
            }
        }
        else {
            throw new DocumentException("Invalid message format");
        }

        Element numeroNode = ordineNode.getChild(pianteName);
        String str = numeroNode.getText();
        if ( str == null || str.length() == 0 ) {
            throw new DocumentException("Invalid message format");
        }
    }
}
```

```

        try{
            numero=Integer.parseInt(str);
        } catch (NumberFormatException e) {
            throw new DocumentException("Invalid message format");
        }
    }
}

public EbxmlMessage toMessage() throws DocumentException {
    Element ordineTransformNode = new Element(ordineTransformName);
    Element dataNode = new Element(dataName);
    dataNode.setText(data);
    ordineTransformNode.addContent(dataNode);
    Element pianteNode = new Element(pianteName);
    pianteNode.setText(Integer.toString(numero));
    ordineTransformNode.addContent(pianteNode);

    Document doc = new Document(ordineTransformNode);
    XMLOutputter fmt = new XMLOutputter();
    ByteArrayOutputStream tempStream = new ByteArrayOutputStream();
    try {
        fmt.output(doc, tempStream);
    }
    catch (IOException e) {
        throw new DocumentException("Unexpected I/O Error occur:" +
e.toString());
    }
    byte[] content = tempStream.toByteArray();
    ByteArrayDataSource source = new ByteArrayDataSource(content,
"text/xml", "haha");
    try {
        EbxmlMessage message = new EbxmlMessage();
        AttachmentDataSource payloadSource = new
AttachmentDataSource(content, "text/xml", "haha");
        message.addPayloadContainer(new DataHandler(payloadSource),
contentId, "description");
        message.addMessageHeader();
        MessageHeader header = message.getMessageHeader();
        header.setCpaId(cpa);
        header.setService(service);
        header.setAction(action);
        return message;
    }
    catch (SOAPException e) {
        throw new DocumentException("SOAPException occur:" + e.toString());
    }
}

public boolean isPackagingReady(StringBuffer validationMessage) {
    if ( data == null || data.length() == 0 ) {
        validationMessage.append("data non valida");
        return false;
    }
    if ( numero <= 0 ) {
        validationMessage.append("numero di piante non valido");
        return false;
    }
    return true;
}

public String getData() {
    return data;
}

```

```
    }

    public int getNumero() {
        return numero;
    }

    public void setData(String tAdd) {
        data = tAdd;
    }

    public void setNumero(int number) {
        numero = number;
    }
}
```

ORDERCONTROLLER.JAVA

```
import hk.hku.cecid.phoenix.ebmail.plugin.Controller;
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentData;
import java.util.Map;
import java.util.HashMap;
import java.awt.Component;

public class OrderController implements Controller {
    private Map properties = new HashMap();

    private OrderDocumentData docData;

    private OrderPane orderPane;

    //costruttore
    public OrderController(OrderDocumentData docData) {
        this.docData = docData;
        orderPane = new OrderPane();
        updateView();
    }

    public Component getView() {
        return orderPane;
    }

    public DocumentData getDocumentData() {
        return docData;
    }

    public void updateView() {
        orderPane.setData(docData.getData());
        orderPane.setNumero(docData.getNumero());
    }

    public void updateData() {
        docData.setData(orderPane.getData());
        docData.setNumero(orderPane.getNumero());
    }

    public void freezeView() {
        orderPane.setEditable(false);
    }

    public void unfreezeView() {
```

```
        orderPane.setEditable(true);
    }

    public String[] getExtendActionNames() {
        return null;
    }

    public void callExtendAction(String actionName) {
        throw new Error("No Extend Actions defined");
    }

    public void setProperty(String key, Object obj) {
        properties.put(key, obj);
    }

    public void clearProperties() {
        properties.clear();
    }
}
```

ORDERDOCUMENTHANDLER.JAVA

```
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentHandler;
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentException;
import hk.hku.cecid.phoenix.ebmail.plugin.Controller;
import hk.hku.cecid.phoenix.ebmail.plugin.DocumentData;

public class OrderDocumentHandler extends DocumentHandler {
    public Controller browseDocument(DocumentData document) throws
    DocumentException {
        if ( document instanceof OrderDocumentData ) {
            OrderDocumentData orderData = (OrderDocumentData)document;
            OrderController orderController = new OrderController(orderData);
            orderController.freezeView();
            return orderController;
        }
        else {
            throw new DocumentException("Invalid Document Data");
        }
    }

    public Controller composeDocument(DocumentData document) throws
    DocumentException {
        if ( document instanceof OrderDocumentData ) {
            OrderDocumentData orderData = (OrderDocumentData)document;
            OrderController orderController = new OrderController(orderData);
            orderController.unfreezeView();
            return orderController;
        }
        else {
            throw new DocumentException("Invalid Document Data");
        }
    }

    public DocumentData[] followUpDocument(DocumentData document) throws
    DocumentException {
        throw new DocumentException("Not Supported yet");
        //DocumentData[] tmp;
        //return tmp;
    }
}
```

BIBLIOGRAFIA

- [1] **Tracciabilità di filiera a garanzia delle produzioni agroalimentare**
“Unioncamere, <http://www.agroqualita.it/TRACCIABILITA.pdf>“
- [2] **Sistema di Rintracciabilità nelle Filiere Agroalimentari – Principi generali per la progettazione e l’attuazione**
“Norma UNI 10939, <http://www.uni.com/it/>“
- [3] **Regolamento CE 178/2002 del parlamento europeo e del consiglio del 28/01/2002**
“http://europa.eu.int/eur-lex/pri/it/oj/dat/2002/l_031/l_03120020201it00010024.pdf“
- [4] **Attuazione delle Dir. 93/43/CEE e 96/3/CE concernenti l’igiene dei prodotti alimentari** “Dec.Lgs. 155/97,
“<http://www.parlamento.it/parlam/leggi/deleghe/97155dl.htm>“
- [5] **Fondamenti dei sistemi di tracciabilità nell’agroalimentare** “Indicod,
”http://www.indicod-ecr.it/agroalimentare/sicurezza/documenti/Fondamenti_tracciabilita.pdf“
- [6] **Manuale operativo delle procedure dei controlli di conformità alle norme comuni di qualità sui prodotti ortofrutticoli**
“<http://www.politicheagricole.it/PRODUZIONE/ORTO/commercializzazione/MANUALE.pdf>”
- [7] **Manuale di corretta prassi per la Produzione integrata del cavolo Broccolo e Verza, Parco Tecnologico Agroalimentare Dell’Umbria, 2001**
<http://www.parco3a.org/pdf/Mcpp%20cavolo.pdf>
- [8] **Disciplinari di produzione per marchio Agriqualità**
<http://www.arsia.toscana.it/agriqualita/disciplinari.htm>
- [9] **Regolamento di Controllo Indicazione Geografica Protetta Lenticchia Di Castelluccio di Norcia**
<http://www.parco3a.org/pdf/RDC01Rev1.zip>
- [10] **Java Mattone Dopo Mattone**
“Massimiliano Tarquini, <http://java-net.tv>“
- [11] **Mokabook**
“<http://www.mokabyte.it>“
- [12] **ebXML Business Process Specification Schema v1.01 6**
“Business Process Project Team, <http://www.ebxml.org/specs/ebBPSS.pdf>“
- [13] **Collaboration-Protocol Profile and Agreement Specification v2.0**
“OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee,
<http://www.ebxml.org/specs/ebcpp-2.0.pdf>“

[14] **Message Service Specification v2.0**

“OASIS ebXML Messaging Services Technical Committee,
<http://www.ebxml.org/specs/ebMS2.pdf>”

[15] **Introduction to ebXML**

“<http://www-128.ibm.com/developerworks/edu/x-dw-xebxml-i.html>”

[16] **Cerere: an information system supporting traceability in the food supply chain**

M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, A. Tomasi

[17] **ebMail: Simple to transact**

“http://www.freebxml.org/ebmail_brochure.pdf”

[18] **ebMail plug-in development guide**

“http://www.freebxml.org/register.php?target_file=ebmail_plugin_develop_guide.zip”

[19] **ebMail plug-in development tutorial**

“http://www.freebxml.org/register.php?target_file=ebmail_plugin_develop_guide.zip”